
NLSY97 CODEBOOK SUPPLEMENT

MAIN FILE ROUND 3

**Prepared for the
U.S. Department of Labor by**

**Center for Human Resource Research
The Ohio State University**

**Under contract with
National Opinion Research Center
University of Chicago**

2001

**NLSY97 Appendix 5:
Income and Assets Variable Creation**

HOUSEHOLD INCOME AND ASSETS

Variables Created: CV_HH_NET_WORTH_Y
CV_INCOME_GROSS_YR
CV_HH_POV_RATIO

Variables Used

| Name in Program | Question Name on CD | Name in Program | Question Name on CD |
|--------------------|---------------------------|--------------------|---------------------------|
| pubid | PUBID | YAS4844L, YAS4844U | YAST-4844-000001, ~000002 |
| YAS50-YAS210 | YAST-050-YAST-210 | YAS4904L, YAS4904U | YAST-4904-000001, ~000002 |
| YAS600 | YAST-600 | YAS5040 | YAST-5040 |
| YAS900 | YAST-900 | YAS5060 | YAST-5060 |
| YAS1400 | YAST-1400 | YAS5062 | YAST-5062 |
| YAS1500 | YAST-1500 | A506601-A506620 | YAST-5066.01-.20 |
| YAS1610 | YAST-1610 | A507401-A507420 | YAST-5074.01-.20 |
| YAS1860-YAS1866 | YAST-1860-YAST-1866 | A508001-A508020 | YAST-5080.01-.20 |
| YAS2520 | YAST-2520 | A508201-A508220 | YAST-5082.01-.20 |
| YAS2550-YAS2556 | YAST-2550-YAST-2556 | A508601, A508619 | YAST-5086.01, .19 |
| YAS2760-YAS2766 | YAST-2760-YAST-2766 | YAS5150 | YAST-5150 |
| YAS3380-YAS3386 | YAST-3380-YAST-3386 | YAS5152 | YAST-5152 |
| YAS3384L, YAS3384U | YAST-3384-000001, ~000002 | A516401-A516420 | YAST-5164.01-.20 |
| YAS3740-YAS3790 | YAST-3740-YAST-3790 | A517001-A517020 | YAST-5170.01-.20 |
| YAS3754L, YAS3754U | YAST-3754-000001, ~000002 | A517201-A517220 | YAST-5172.01-.20 |
| YAS3810-YAS3922 | YAST-3810-YAST-3922 | A517601 | YAST-5176.01 |
| YAS3924L, YAS3924U | YAST-3924-000001, ~000002 | YAS5210-YAS5226 | YAST-5210-YAST-5226 |
| YAS3950-YAS3966 | YAST-3950-YAST-3966 | YAS5224L, YAS5224U | YAST-5224-000001, ~000002 |
| YAS3964L, YAS3964U | YAST-3964-000001, ~000002 | HIRELY01-HIRELY14 | HHI_RELY.01-14 |
| YAS4010-YAS4036 | YAST-4010-YAST-4036 | YI_300 | YINC-300 |
| YAS4034L, YAS4034U | YAST-4034-000001, ~000002 | YI_400 | YINC-400 |
| YAS4140-YAS4162 | YAST-4140-YAST-4162 | YI_1400-YI_3400 | YINC-1400-YINC-3400 |
| YAS4270-YAS4296 | YAST-4270-YAST-4296 | YI_3900-YI_5300 | YINC-3900-YINC-5300 |
| YAS4294L, YAS4294U | YAST-4294-000001, ~000002 | YI_55001-YI_55007 | YINC-5500-000000-~000006 |
| YAS4400-YAS4426 | YAST-4400-YAST-4426 | YI_5600-YI_9900 | YINC-5600-YINC-9900 |
| YAS4424L, YAS4424U | YAST-4424-000001, ~000002 | YI_10000-YI_10800 | YINC-10000-YINC-10800 |
| YAS4530-YAS4556 | YAST-4530-YAST-4556 | I1110001-I1110014 | YINC-11100.01-.14 |
| YAS4554L, YAS4554U | YAST-4554-000001, ~000002 | I1160001-I1160013 | YINC-11600.01-.13 |
| YAS4660-YAS4686 | YAST-4660-YAST-4686 | I1170001-I1170013 | YINC-11700.01-.13 |
| YAS4684L, YAS4684U | YAST-4684-000001, ~000002 | hhszie | CV_HH_SIZE |
| YAS4790-YAS4906 | YAST-4790-YAST-4906 | under18 | CV_HH_UNDER_18 |
| YAS4814L, YAS4814U | YAST-4814-000001, ~000002 | | |

This program creates the household net worth and gross household income variables. The household net worth variable is an actual number that results from adding the values of all assets and subtracting liabilities of the household. The gross household income variable includes total annual cash receipts before taxes from all sources. The program then creates a ratio comparing the household's total income to federal poverty guidelines based on the number of household residents and the number of members under age 18.

Researchers should note that, like many income and asset variables in the data set, these three variables are topcoded to protect respondent privacy. More information about topcoding is available in the *NLSY97 User's Guide*.

******* SECTION 1: GROSS HOUSEHOLD INCOME *******

```
/* In income section, flag top three items involved the most respondents: I. nonfarm employment income -- flag1400
II. father's income -- flag8700 III. mother's income -- flag9200
flag=0 indicating accurate income reported; flag=1 indicating estimated income reported */
```

```
/*Create components of the hh net worth and gross income, such as income from wages and salaries, interests, etc. */
```

```

/* Income from non-farm employment (nfarmwgY) */
nfarmwgY=-4;
flag1400=0;
if (YI_1400=1 and YI_1700 ne -1 and YI_1700 ne -2 and YI_1700 ne -3)
    or (YI_1400=-1 and YI_1600=1 and YI_1700 ne -1 and YI_1700 ne -2 and YI_1700 ne -3)
    or (YI_1400=-2 and YI_1500=1 and YI_1700 ne -1 and YI_1700 ne -2 and YI_1700 ne -3)
    or (YI_1400=-2 and YI_1500=-1 and YI_1600=1 and YI_1700 ne -1 and YI_1700 ne -2 and YI_1700 ne -3)
    then nfarmwgY=YI_1700;
if (YI_1400=1 or (YI_1400=-1 and YI_1600=1) or (YI_1400=-2 and YI_1500=1)
    or (YI_1400=-2 and YI_1500=-1 and YI_1600=1) or (YI_1400=-3 and YI_1600=1)
    or (YI_1400=-3 and YI_1500=1) or (YI_1400=-2 and YI_1500=-3 and YI_1600=1))
    and (YI_1700 eq -1 or YI_1700 eq -2 or YI_1700 eq -3) then do;
    if YI_1800=1 then do; nfarmwgY=2500; flag1400=1; end;
    if YI_1800=2 then do; nfarmwgY=7500; flag1400=1; end;
    if YI_1800=3 then do; nfarmwgY=17500; flag1400=1; end;
    if YI_1800=4 then do; nfarmwgY=37500; flag1400=1; end;
    if YI_1800=5 then do; nfarmwgY=75000; flag1400=1; end;
    if YI_1800=6 then do; nfarmwgY=175000; flag1400=1; end;
    if YI_1800=7 then do; nfarmwgY=250001; flag1400=1; end;
end;
if YI_1400=0 or YI_1500=0 or YI_1600=0 then nfarmwgY=0;
if YI_1600=-1 or YI_1800=-1 then nfarmwgY=-1;
if YI_1500=-2 or YI_1800=-2 then nfarmwgY=-2;
if YI_1600=-3 or YI_1500=-3 or YI_1800=-3 then nfarmwgY=-3;

/* For all the questions below, the youth must be INDEPENDENT (YI_1900=1) */
/* Income from farm and self-employment (farmwgY)*/
farmwgY=-4;
if YI_2000=1 and YI_2100 ne -1 and YI_2100 ne -2 and YI_2100 ne -3 then farmwgY=YI_2100;
if YI_2000=1 and (YI_2100 eq -1 or YI_2100 eq -2 or YI_2100 eq -3) then do;
    if YI_2200=1 then do; farmwgY=-2; end;           if YI_2200=2 then do; farmwgY=2500; end;
    if YI_2200=3 then do; farmwgY=7500; end;         if YI_2200=4 then do; farmwgY=17500; end;
    if YI_2200=5 then do; farmwgY=37500; end;        if YI_2200=6 then do; farmwgY=75000; end;
    if YI_2200=7 then do; farmwgY=175000; end;       if YI_2200=8 then do; farmwgY=250001; end;
end;
if YI_2000=0 then farmwgY=0;
if YI_2000=-1 or YI_2200=-1 then farmwgY=-1;
if YI_2000=-2 or YI_2200=-2 then farmwgY=-2;
if YI_2000=-3 or YI_2200=-3 then farmwgY=-3;

/* Non-farm income from the spouse/partner (nfarmwgP and farmwgP) */
nfarmwgP=-4;
if (YI_2300=1 and YI_2400=1 and YI_2600 ne -1 and YI_2600 ne -2 and YI_2600 ne -3) or
    (YI_2300=1 and YI_2400=-1 and YI_2500=1 and YI_2600 ne -1 and YI_2600 ne -2 and YI_2600 ne -3)
    then nfarmwgP=YI_2600;
if (YI_2300=1 and YI_2400=1 and (YI_2600 eq -1 or YI_2600 eq -2 or YI_2600 eq -3)) or
    (YI_2300=1 and YI_2400=-1 and YI_2500=1 and (YI_2600 eq -1 or YI_2600 eq -2 or YI_2600 eq -3)) then do;
    if YI_2700=1 then do; nfarmwgP=2500; end;           if YI_2700=2 then do; nfarmwgP=7500; end;
    if YI_2700=3 then do; nfarmwgP=17500; end;          if YI_2700=4 then do; nfarmwgP=37500; end;
    if YI_2700=5 then do; nfarmwgP=75000; end;          if YI_2700=6 then do; nfarmwgP=175000; end;
    if YI_2700=7 then do; nfarmwgP=250001; end;
end;
if YI_2300=0 or YI_2400=0 or YI_2500=0 then nfarmwgP=0;
if YI_2300=-1 or YI_2500=-1 or YI_2700=-1 then nfarmwgP=-1;
if YI_2300=-2 or YI_2400=-2 or YI_2500=-2 or YI_2700=-2 then nfarmwgP=-2;
if YI_2300=-3 or YI_2400=-3 or YI_2500=-3 or YI_2700=-3 then nfarmwgP=-3;

```

```

/* Farm-related income from the spouse/partner */
farmwgP=-4;
if YI_2300=1 and YI_2900=1 and YI_3000 ne -1 and YI_3000 ne -2 and YI_3000 ne -3 then farmwgP=YI_3000;
if YI_2300=1 and YI_2900=1 and (YI_3000 eq -1 or YI_3000 eq -2 or YI_3000 eq -3) then do;
    if YI_3100=1 then do; farmwgP=-2; end;           if YI_3100=2 then do; farmwgP=2500; end;
    if YI_3100=3 then do; farmwgP=7500; end;         if YI_3100=4 then do; farmwgP=17500; end;
    if YI_3100=5 then do; farmwgP=37500; end;        if YI_3100=6 then do; farmwgP=75000; end;
    if YI_3100=7 then do; farmwgP=175000; end;       if YI_3100=8 then do; farmwgP=250001; end;
end;
if YI_2300=0 or YI_2900=0 then farmwgP=0;
if YI_2300=-1 or YI_2900=-1 or YI_3100=-1 then farmwgP=-1;
if YI_2300=-2 or YI_2900=-2 or YI_3100=-2 then farmwgP=-2;
if YI_2300=-3 or YI_2900=-3 or YI_3100=-3 then farmwgP=-3;

/* Child support payment (childsuY) */
childsuY=-4;
if YI_3300=1 and YI_3900=1 and YI_4000=1 and YI_4100 ne -1 and YI_4100 ne -2 and YI_4100 ne -3
    then childsuY=YI_4100;
if YI_3300=1 and YI_3900=1 and YI_4000=1 and (YI_4100 eq -1 or YI_4100 eq -2 or YI_4100 eq -3) then do;
    if YI_4200=1 then do; childsuY=500; end;           if YI_4200=2 then do; childsuY=1750; end;
    if YI_4200=3 then do; childsuY=3750; end;         if YI_4200=4 then do; childsuY=7500; end;
    if YI_4200=5 then do; childsuY=17500; end;        if YI_4200=6 then do; childsuY=37500; end;
    if YI_4200=7 then do; childsuY=50001; end;
end;
if YI_3300=0 or YI_3900=0 or YI_4000=0 then childsuY=0;
if YI_3300=-1 or YI_3900=-1 or YI_4000=-1 or YI_4200=-1 then childsuY=-1;
if YI_3300=-2 or YI_3900=-2 or YI_4000=-2 or YI_4200=-2 then childsuY=-2;
if YI_3300=-1 or YI_3900=-1 or YI_4000=-3 or YI_4100=-3 or YI_4200=-3 then childsuY=-3;

/* Interest payments received by the youth (Y) and his/her partner/spouse */
interesY=-4;
if YI_4300=1 and YI_4400 ne -1 and YI_4400 ne -2 and YI_4400 ne -3 then interesY=YI_4400;
if YI_4300=1 and (YI_4400 eq -1 or YI_4400 eq -2 or YI_4400 eq -3) then do;
    if YI_4500=1 then do; interesY=250; end;           if YI_4500=2 then do; interesY=750; end;
    if YI_4500=3 then do; interesY=1750; end;         if YI_4500=4 then do; interesY=3750; end;
    if YI_4500=5 then do; interesY=6250; end;         if YI_4500=6 then do; interesY=8750; end;
    if YI_4500=7 then do; interesY=10001; end;
end;
if YI_4300=0 then interesY=0;
if YI_4300=-1 or YI_4500=-1 then interesY=-1;
if YI_4300=-2 or YI_4500=-2 then interesY=-2;
if YI_4300=-3 or YI_4400=-3 or YI_4500=-3 then interesY=-3;

/* Dividends from stocks and mutual funds */
dividend=-4;
if YI_4600=1 and YI_4700 ne -1 and YI_4700 ne -2 and YI_4700 ne -3 then dividend=YI_4700;
if YI_4600=1 and (YI_4700 eq -1 or YI_4700 eq -2 or YI_4700 eq -3) then do;
    if YI_4800=1 then do; dividend=250; end;           if YI_4800=2 then do; dividend=750; end;
    if YI_4800=3 then do; dividend=1750; end;         if YI_4800=4 then do; dividend=3750; end;
    if YI_4800=5 then do; dividend=6250; end;         if YI_4800=6 then do; dividend=8750; end;
    if YI_4800=7 then do; dividend=10001; end;
end;
if YI_4600=0 then dividend=0;
if YI_4600=-1 or YI_4800=-1 then dividend=-1;
if YI_4600=-2 or YI_4800=-2 then dividend=-2;

```

```

if YI_4600=-3 or YI_4700=-3 or YI_4800=-3 then dividend=-3;

/* Rental income */
rentalIY=-4;
if YI_4900=1 and YI_5000 ne -1 and YI_5000 ne -2 and YI_5000 ne -3 then rentalIY=YI_5000;
if YI_4900=1 and (YI_5000 eq -1 or YI_5000 eq -2 or YI_5000 eq -3) then do;
    if YI_5100=1 then do; rentalIY=500; end;           if YI_5100=2 then do; rentalIY=1750; end;
    if YI_5100=3 then do; rentalIY=3750; end;         if YI_5100=4 then do; rentalIY=7500; end;
    if YI_5100=5 then do; rentalIY=17500; end;        if YI_5100=6 then do; rentalIY=37500; end;
    if YI_5100=7 then do; rentalIY=50001; end;
end;
if YI_4900=0 then rentalIY=0;
if YI_4900=-1 or YI_5100=-1 then rentalIY=-1;
if YI_4900=-2 or YI_5100=-2 then rentalIY=-2;
if YI_4900=-3 or YI_5000=-3 or YI_5100=-3 then rentalIY=-3;

/* Property or money from estates, trusts, annuities or inheritances */
estatesY=-4;
if YI_5200=1 and YI_5300 ne -1 and YI_5300 ne -2 and YI_5300 ne -3 then estatesY=YI_5300;
if YI_5200=1 and (YI_5300 eq -1 or YI_5300 eq -2 or YI_5300 eq -3) then do;
    if YI_5400=1 then do; estatesY=2500; end;           if YI_5400=2 then do; estatesY=7500; end;
    if YI_5400=3 then do; estatesY=17500; end;         if YI_5400=4 then do; estatesY=37500; end;
    if YI_5400=5 then do; estatesY=75000; end;        if YI_5400=6 then do; estatesY=175000; end;
    if YI_5400=7 then do; estatesY=250001; end;
end;
if YI_5200=0 then estatesY=0;
if YI_5200=-1 or YI_5400=-1 then estatesY=-1;
if YI_5200=-2 or YI_5400=-2 then estatesY=-2;
if YI_5200=-3 or YI_5300=-3 or YI_5400=-3 then estatesY=-3;

/* Income from other sources: SS payments, pension and retirement income, alimony,
payments from insurance policies, etc... */
pensionY=-4;
if YI_7600=1 and YI_7700 ne -1 and YI_7700 ne -2 and YI_7700 ne -3 then pensionY=YI_7700;
if YI_7600=1 and (YI_7700 eq -1 or YI_7700 eq -2 or YI_7700 eq -3) then do;
    if YI_7800=1 then do; pensionY=500; end;           if YI_7800=2 then do; pensionY=1750; end;
    if YI_7800=3 then do; pensionY=3750; end;         if YI_7800=4 then do; pensionY=7500; end;
    if YI_7800=5 then do; pensionY=17500; end;        if YI_7800=6 then do; pensionY=37500; end;
    if YI_7800=7 then do; pensionY=50001; end;
end;
if YI_7600=0 then pensionY=0;
if YI_7600=-1 or YI_7800=-1 then pensionY=-1;
if YI_7600=-2 or YI_7800=-2 then pensionY=-2;
if YI_7600=-3 or YI_7700=-3 or YI_7800=-3 then pensionY=-3;

/* If the youth lived with the father, the father's income */
flag8700=0; faincome=-4;
/* identifying the father is in the household */
if (hirely01=4 or hirely02=4 or hirely03=4 or hirely04=4 or hirely05=4 or hirely06=4 or hirely07=4 or
    hirely08=4 or hirely09=4 or hirely10=4 or hirely11=4 or hirely12=4 or hirely13=4 or hirely14=4) then do;
if YI_8700=1 and YI_8800 ne -1 and YI_8800 ne -2 and YI_8800 ne -3 then faincome=YI_8800;
if YI_8700=1 and (YI_8800 eq -1 or YI_8800 eq -2 or YI_8800 eq -3) then do;
    if YI_8900=1 then do; faincome=2500; flag8700=1; end;
    if YI_8900=2 then do; faincome=7500; flag8700=1; end;
    if YI_8900=3 then do; faincome=17500; flag8700=1; end;
    if YI_8900=4 then do; faincome=37500; flag8700=1; end;

```

```

if YI_8900=5 then do; faincome=75000; flag8700=1; end;
if YI_8900=6 then do; faincome=175000; flag8700=1; end;
if YI_8900=7 then do; faincome=250001; flag8700=1; end;
end;
if YI_8700=0 then faincome=0;
if YI_8700=-1 or YI_8900=-1 then faincome=-1;
if YI_8700=-2 or YI_8900=-2 then faincome=-2;
if YI_8700=-3 or YI_8800=-3 or YI_8900=-3 then faincome=-3;
end;

/* If the youth lived with the mother, the mother's income */
flag9200=0; maincome=-4;
/* identify the mother is in the household */
if (hirely01=3 or hirely02=3 or hirely03=3 or hirely04=3 or hirely05=3 or hirely06=3 or hirely07=3 or
    hirely08=3 or hirely09=3 or hirely10=3 or hirely11=3 or hirely12=3 or hirely13=3 or hirely14=3) then do;
if YI_9200=1 and YI_9300 ne -1 and YI_9300 ne -2 and YI_9300 ne -3 then maincome=YI_9300;
if YI_9200=1 and (YI_9300 eq -1 or YI_9300 eq -2 or YI_9300 eq -3) then do;
    if YI_9400=1 then do; maincome=2500; flag9200=1; end;
    if YI_9400=2 then do; maincome=7500; flag9200=1; end;
    if YI_9400=3 then do; maincome=17500; flag9200=1; end;
    if YI_9400=4 then do; maincome=37500; flag9200=1; end;
    if YI_9400=5 then do; maincome=75000; flag9200=1; end;
    if YI_9400=6 then do; maincome=175000; flag9200=1; end;
    if YI_9400=7 then do; maincome=250001; flag9200=1; end;
end;
if YI_9200=0 then maincome=0;
if YI_9200=-1 or YI_9400=-1 then maincome=-1;
if YI_9200=-2 or YI_9400=-2 then maincome=-2;
if YI_9200=-3 or YI_9400=-3 then maincome=-3;
end;

/* If the youth lives with the male guardian */
mgincome=-4;
if YI_9600=1 and YI_9700=1 and YI_9800 ne -1 and YI_9800 ne -2 and YI_9800 ne -3 then mgincome=YI_9800;
if YI_9600=1 and YI_9700=1 and (YI_9800 eq -1 or YI_9800 eq -2 or YI_9800 eq -3) then do;
    if YI_9900=1 then do; mgincome=2500; end;           if YI_9900=2 then do; mgincome=7500; end;
    if YI_9900=3 then do; mgincome=17500; end;         if YI_9900=4 then do; mgincome=37500; end;
    if YI_9900=5 then do; mgincome=75000; end;         if YI_9900=6 then do; mgincome=175000; end;
    if YI_9900=7 then do; mgincome=250001; end;
end;
if YI_9700=0 then mgincome=0;
if YI_9600=-1 or YI_9700=-1 or YI_9900=-1 then mgincome=-1;
if YI_9600=-2 or YI_9700=-2 or YI_9900=-2 then mgincome=-2;
if YI_9600=-3 or YI_9700=-3 or YI_9800=-3 or YI_9900=-3 then mgincome=-3;

/* If the youth lives with female guardian */
fgincome=-4;
if YI_10100=1 and YI_10200=1 and YI_10300 ne -1 and YI_10300 ne -2 and YI_10300 ne -3
    then fgincome=YI_10300;
if YI_10100=1 and YI_10200=1 and (YI_10300 eq -1 or YI_10300 eq -2 or YI_10300 eq -3) then do;
    if YI_10400=1 then do; fgincome=2500; end;           if YI_10400=2 then do; fgincome=7500; end;
    if YI_10400=3 then do; fgincome=17500; end;         if YI_10400=4 then do; fgincome=37500; end;
    if YI_10400=5 then do; fgincome=75000; end;         if YI_10400=6 then do; fgincome=175000; end;
    if YI_10400=7 then do; fgincome=250001; end;
end;
if YI_10200=0 then fgincome=0;

```

```

if YI_10100=-1 or YI_10200=-1 or YI_10400=-1 then fgincome=-1;
if YI_10100=-2 or YI_10200=-2 or YI_10400=-2 then fgincome=-2;
if YI_10100=-3 or YI_10200=-3 or YI_10300=-3 or YI_10400=-3 then fgincome=-3;

/* Income from household members of 14 or older other than spouse/partner */
array otfamI otfamI01-otfamI14;
array I11100 I1110001-I1110014;
array I11600 I1160001-I1160014;
array I11700 I1170001-I1170014;
do I=1 to 14;
otfamI(I)=-4;
if YI_10800=1 then do;
  if I11100(I)=0 and I11600(I) ne -1 and I11600(I) ne -2 and I11600(I) ne -3 and I11600(I) ne -4
    then otfamI(I)=I11600(I);
  if I11100(I)=0 and (I11600(I) eq -1 or I11600(I) eq -2 or I11600(I) eq -3) then do;
    if I11700(I)=1 then do; otfamI(I)=2500; end;           if I11700(I)=2 then do; otfamI(I)=7500; end;
    if I11700(I)=3 then do; otfamI(I)=17500; end;          if I11700(I)=4 then do; otfamI(I)=37500; end;
    if I11700(I)=5 then do; otfamI(I)=75000; end;          if I11700(I)=6 then do; otfamI(I)=175000; end;
    if I11700(I)=7 then do; otfamI(I)=250001; end;
  end;
  if I11100(I)=-1 or I11700(I)=-1 then otfamI(I)=-1;
  if I11100(I)=-2 or I11700(I)=-2 then otfamI(I)=-2;
  if I11100(I)=-3 or I11600(I)=-3 or I11700(I)=-3 then otfamI(I)=-3;
end;
end;

/* We now create gross hh income according to the youth*/
groshhIY=0;
if YI_1900=1 then do;
  if nfarmwgY not in (-1,-2,-3,-4) then groshhIY=groshhIY+nfarmwgY;
  if farmwgY not in (-1,-2,-3,-4) then groshhIY=groshhIY+farmwgY;
  if nfarmwgP not in (-1,-2,-3,-4) then groshhIY=groshhIY+nfarmwgP;
  if farmwgP not in (-1,-2,-3,-4) then groshhIY=groshhIY+farmwgP;
  if childsuY not in (-1,-2,-3,-4) then groshhIY=groshhIY+childsuY;
  if interesY not in (-1,-2,-3,-4) then groshhIY=groshhIY+interesY;
  if dividend not in (-1,-2,-3,-4) then groshhIY=groshhIY+dividend;
  if rentalIY not in (-1,-2,-3,-4) then groshhIY=groshhIY+rentalIY;
  if estatesY not in (-1,-2,-3,-4) then groshhIY=groshhIY+estatesY;
  if pensionY not in (-1,-2,-3,-4) then groshhIY=groshhIY+pensionY;
  if faincome not in (-1,-2,-3,-4) then groshhIY=groshhIY+faincome;
  if maincome not in (-1,-2,-3,-4) then groshhIY=groshhIY+maincome;
  if mgincome not in (-1,-2,-3,-4) then groshhIY=groshhIY+mgincome;
  if fgincome not in (-1,-2,-3,-4) then groshhIY=groshhIY+fgincome;
  if prgamt not in (-1,-2,-3,-4) then groshhIY=groshhIY+prgamt;
  do I=1 to 14; if otfamI[I] not in (-1,-2,-3,-4) then groshhIY=groshhIY+otfamI[I]; end;
end;

if YI_1900=-1 or YI_8500=-1 or (nfarmwgY=-1 or farmwgY=-1 or nfarmwgP=-1 or farmwgP=-1 or childsuY=-1 or
  interesY=-1 or dividend=-1 or rentalIY=-1 or estatesY=-1 or pensionY=-1 or faincome=-1 or maincome=-1
  or mgincome=-1 or fgincome=-1 or otfamI01=-1 or otfamI02=-1 or otfamI03=-1 or otfamI04=-1 or
  otfamI04=-1 or otfamI05=-1 or otfamI06=-1 or otfamI07=-1 or otfamI08=-1 or otfamI09=-1 or otfamI10=-1
  or otfamI11=-1 or otfamI12=-1 or otfamI13=-1 or otfamI14=-1 or prgamt=-1) then groshhIY=-1;
if YI_1900=-2 or YI_8500=-2 or (nfarmwgY=-2 or farmwgY=-2 or nfarmwgP=-2 or farmwgP=-2 or childsuY=-2 or
  interesY=-2 or dividend=-2 or rentalIY=-2 or pensionY=-2 or estatesY=-2 or faincome=-2 or maincome=-2
  or mgincome=-2 or fgincome=-2 or otfamI01=-2 or otfamI02=-2 or otfamI03=-2 or otfamI04=-2 or

```

```

otfamI04=-2 or otfamI05=-2 or otfamI06=-2 or otfamI07=-2 or otfamI08=-2 or otfamI09=-2 or otfamI10=-2
or otfamI11=-2 or otfamI12=-2 or otfamI13=-2 or otfamI14=-2 or prgamt=-2) then groshhIY=-2;
if YI_1900=-3 or YI_8500=-3 or (nfarmwgY=-3 or farmwgY=-3 or nfarmwgP=-3 or farmwgP=-3 or childsuY=-3 or
interesY=-3 or dividend=-3 or rentalIY=-3 or pensionY=-3 or estatesY=-3 or faincome=-3 or maincome=-3
or mgincome=-3 or fgincome=-3 or otfamI01=-3 or otfamI02=-3 or otfamI03=-3 or otfamI04=-3 or
otfamI04=-3 or otfamI05=-3 or otfamI06=-3 or otfamI07=-3 or otfamI08=-3 or otfamI09=-3 or otfamI10=-3
or otfamI11=-3 or otfamI12=-3 or otfamI13=-3 or otfamI14=-3 or prgamt=-3) then groshhIY=-3;
if YI_1900=0 or YI_1900=-4 then groshhIY=-4;
if YAS50=-5 then groshhIY=-5; /*give the people who were not interviewed in round 3 value -5.*/

```

/*** SECTION 2: HOUSEHOLD NET WORTH *****/**

```

/* In asset section, flag top three items involved the most respondents: I. savings -- flag4400
II. present value of vehicles owned -- flag4790 III. money still owed on these vehicles -- flag4840
flag=0 indicating accurate value reported; flag=1 indicating estimated value reported */

/* If the youth OWNS some land, its present value. */
pvranch=0;
if YAS1400=1 then do;
  if YAS1610=1 or YAS1610=2 then do; /* If the youth reported owning all of it */
    if YAS1860=1 then pvranch=YAS1862;
    if YAS1860=2 then pvranch=YAS1864L+(YAS1864U-YAS1864L)/2;
    if ((YAS1860 ne 1 and YAS1860 ne 2) or (YAS1862=-1 or YAS1862=-2)) then do;
      if YAS1866=1 then do; pvranch=12500; end; if YAS1866=2 then do; pvranch=37500; end;
      if YAS1866=3 then do; pvranch=75000; end; if YAS1866=4 then do; pvranch=175000;
    end;
    if YAS1866=5 then do; pvranch=375000; end; if YAS1866=6 then do; pvranch=750000; end;
    if YAS1866=7 then do; pvranch=1000001; end;
  end;
end;
if (YAS1610=3 or YAS1610=4) and YAS2120=100 then do; /* If the youth reported owning only part of it */
  if YAS2140=1 then pvranch=YAS2142;
  if YAS2140=2 then pvranch=(YAS2144L+(YAS2144U-YAS2144L)/2);
  if ((YAS2140 ne 1 and YAS2140 ne 2) or (YAS2142=-1 or YAS2142=-2)) then do;
    if YAS2146=1 then do; pvranch=12500; end; if YAS2146=2 then do; pvranch=37500; end;
    if YAS2146=3 then do; pvranch=75000; end; if YAS2146=4 then do; pvranch=175000;
  end;
  if YAS2146=5 then do; pvranch=375000; end; if YAS2146=6 then do; pvranch=750000; end;
  if YAS2146=7 then do; pvranch=1000001; end;
end;
if (YAS1610=3 or YAS1610=4) and YAS2120 ne 100 then do;
  if YAS2170=1 then pvranch=YAS2172;
  if YAS2170=2 then pvranch=(YAS2174L+(YAS2174U-YAS2174L)/2);
  if ((YAS2170 ne 1 and YAS2170 ne 2) or (YAS2172=-1 or YAS2172=-2)) then do;
    if YAS2176=1 then do; pvranch=12500; end; if YAS2176=2 then do; pvranch=37500; end;
    if YAS2176=3 then do; pvranch=75000; end; if YAS2176=4 then do; pvranch=175000;
  end;
  if YAS2176=5 then do; pvranch=375000; end; if YAS2176=6 then do; pvranch=750000; end;
  if YAS2176=7 then do; pvranch=1000001; end;
end;
if YAS1400=-1 or YAS1610=-1 or YAS1864L=-1 or YAS1864U=-1 or YAS1866=-1 or YAS2144L=-1 or
YAS2144U=-1 or YAS2146=-1 or YAS2174L=-1 or YAS2174U=-1 or YAS2176=-1 then pvranch=-1;

```

```

if YAS1400=-2 or YAS1610=-2 or YAS1864L=-2 or YAS1864U=-2 or YAS1866=-2 or YAS2144L=-2 or
    YAS2144U=-2 or YAS2146=-2 or YAS2174L=-2 or YAS2174U=-2 or YAS2176=-2 then pvrank=-2;
if YAS1400=-3 or YAS1610=-3 or YAS1864L=-3 or YAS1864U=-3 or YAS1866=-3 or YAS2144L=-3 or
    YAS2144U=-3 or YAS2146=-3 or YAS2174=-3 or YAS2176=-3 then pvrank=-3;

/* If the youth owns the mobile home and the site, its PRESENT VALUE.*/
pvmbst=0;
if YAS1500=1 then do;
    if (YAS2520=1 or YAS2520=2) and YAS2550=1 then pvmbst=YAS2552;
    if (YAS2520=1 or YAS2520=2) and YAS2550=2 then pvmbst=YAS2554L+(YAS2554U-YAS2554L)/2;
    if (YAS2520=1 or YAS2520=2) and ((YAS2550 ne 1 and YAS2550 ne 2) or (YAS2552=-1 or YAS2552=-2))
then do;
    if YAS2556=1 then do; pvmbst=500; end;           if YAS2556=2 then do; pvmbst=1750; end;
    if YAS2556=3 then do; pvmbst=3750; end;         if YAS2556=4 then do; pvmbst=7500; end;
    if YAS2556=5 then do; pvmbst=17500; end;        if YAS2556=6 then do; pvmbst=37500; end;
    if YAS2556=7 then do; pvmbst=50001; end;
end;
end;
if YAS1500=-1 or YAS2520=-1 or YAS2554L=-1 or YAS2554U=-1 or YAS2556=-1 then pvmbst=-1;
if YAS1500=-2 or YAS2520=-2 or YAS2554L=-2 or YAS2554U=-2 or YAS2556=-2 then pvmbst=-2;
if YAS1500=-3 or YAS2520=-3 or YAS2552=-3 or YAS2554L=-3 or YAS2554U=-3 or YAS2556=-3 then
pvmbst=-3;

/* If the youth owns only the mobile home, its PRESENT VALUE. */
pvmb=0;
if YAS1500=1 then do;
    if (YAS2520=3 or YAS2520=4) and YAS2760>0 then pvmb=YAS2760;
    if (YAS2520=3 or YAS2520=4) and (YAS2764L>-4 and YAS2764L>-4 and YAS2764L ne .)
        then pvmb=(YAS2764L+(YAS2764U-YAS2764L)/2);
    if (YAS2520=3 or YAS2520=4) and (YAS2760=-1 or YAS2760=-2 or YAS2960=-3) then do;
        if YAS2766=1 then do; pvmb=12500; end;           if YAS2766=2 then do; pvmb=37500; end;
        if YAS2766=3 then do; pvmb=75000; end;         if YAS2766=4 then do; pvmb=175000; end;
        if YAS2766=5 then do; pvmb=375000; end;        if YAS2766=6 then do; pvmb=750000; end;
        if YAS2766=7 then do; pvmb=1000001; end;
    end;
end;
if YAS1500=-1 or YAS2520=-1 or YAS2764L=-1 or YAS2764U=-1 or YAS2766=-1 then pvmb=-1;
if YAS1500=-2 or YAS2520=-2 or YAS2764L=-2 or YAS2764U=-2 or YAS2766=-2 then pvmb=-2;
if YAS1500=-3 or YAS2520=-3 or YAS2762=-3 or YAS2764L=-3 or YAS2764U=-3 or YAS2766=-3 then pvmb=-3;

/* If the youth owns only the mobile home site, its PRESENT VALUE.*/
pvst=0;
if YAS1500=1 then do;
    if (YAS2520=5 or YAS2520=6) and YAS3010=1 then pvst=YAS3012;
    if (YAS2520=5 or YAS2520=6) and YAS3010=2 then pvst=(YAS3014L+(YAS3014U-YAS3014L)/2);
    if (YAS2520=5 or YAS2520=6) and ((YAS3010 ne 1 and YAS3010 ne 2) or (YAS3012=-1 or YAS3012=-2))
then do;
    if YAS3016=1 then do; pvst=12500; end;           if YAS3016=2 then do; pvst=57500; end;
    if YAS3016=3 then do; pvst=75000; end;         if YAS3016=4 then do; pvst=175000; end;
    if YAS3016=5 then do; pvst=575000; end;        if YAS3016=6 then do; pvst=750000; end;
    if YAS3016=7 then do; pvst=1000001; end;
end;
end;
if YAS2520=-1 or YAS3014L=-1 or YAS3014U=-1 or YAS3016=-1 or YAS1500=-1 then pvst=-1;
if YAS2520=-2 or YAS3014L=-2 or YAS3014U=-2 or YAS3016=-2 or YAS1500=-2 then pvst=-2;

```

```

if YAS2520=-3 or YAS3012=-3 or YAS3014L=-3 or YAS3014U=-3 or YAS3016=-3 or YAS1500=-3 then pvst=-3;

/* If the respondent owns the apartment, its value */
pvapthm=0;
if YAS3310=1 or YAS3310=2 then do;
  if YAS3380=1 then pvapthm=YAS3382;
  if YAS3380=2 then pvapthm=(YAS3384L+(YAS3384U-YAS3384L)/2);
  if ((YAS3380 ne 1 and YAS3380 ne 2) or (YAS3382=-1 or YAS3382=-2)) then do;
    if YAS3386=1 then do; pvapthm=500; end;           if YAS3386=2 then do; pvapthm=1750; end;
    if YAS3386=3 then do; pvapthm=3750; end;         if YAS3386=4 then do; pvapthm=7500; end;
    if YAS3386=5 then do; pvapthm=17500; end;        if YAS3386=6 then do; pvapthm=37500; end;
    if YAS3386=7 then do; pvapthm=50001; end;
  end;
end;
if YAS3310=-1 or YAS3384L=-1 or YAS3384U=-1 or YAS3386=-1 then pvapthm=-1;
if YAS3310=-2 or YAS3384L=-2 or YAS3384U=-2 or YAS3386=-2 then pvapthm=-2;
if YAS3310=-3 or YAS3382=-3 or YAS3384L=-3 or YAS3384U=-3 or YAS3386=-3 then pvapthm=-3;

/* Mortgage or land contract on land or property */
mortgagY=0;
if YAS3740=1 or YAS3740=2 then do;
  if YAS3750=1 and YAS3752 ge 0 then mortgagY=YAS3752;
  if YAS3750=2 and YAS3754U ge 0 and YAS3754L ge 0 then mortgagY=(YAS3754L+(YAS3754U-
YAS3754L)/2);
  if ((YAS3750 ne 1 and YAS3750 ne 2) or (YAS3752=-1 or YAS3752=-2)) then do;
    if YAS3756=1 then do; mortgagY=500; end;           if YAS3756=2 then do; mortgagY=1750; end;
    if YAS3756=3 then do; mortgagY=3750; end;         if YAS3756=4 then do; mortgagY=7500; end;
    if YAS3756=5 then do; mortgagY=17500; end;        if YAS3756=6 then do; mortgagY=37500; end;
    if YAS3756=7 then do; mortgagY=50001; end;
  end;
end;
if YAS3740=-1 or YAS3754L=-1 or YAS3754U=-1 or YAS3756=-1 then mortgagY=-1;
if YAS3740=-2 or YAS3754L=-2 or YAS3754U=-2 or YAS3756=-2 then mortgagY=-2;
if YAS3740=-3 or YAS3752=-3 or YAS3754L=-3 or YAS3754U=-3 or YAS3756=-3 then mortgagY=-3;

/* Loans from relatives or friends when respondent bought or remodeled the residence */
loanowed=0;
if YAS3790=1 then do;
  if YAS3810=1 and YAS3812 ge 0 then loanowed=YAS3812;
  if YAS3810=2 and YAS3814U ge 0 and YAS3814L ge 0 then loanowed=(YAS3814L+(YAS3814U-
YAS3814L)/2);
  if ((YAS3810 ne 1 and YAS3810 ne 2) or (YAS3812=-1 or YAS3812=-2)) then do;
    if YAS3816=1 then do; loanowed=12500; end;           if YAS3816=2 then do; loanowed=37500; end;
    if YAS3816=3 then do; loanowed=75000; end;         if YAS3816=4 then do; loanowed=175000; end;
    if YAS3816=5 then do; loanowed=375000; end;        if YAS3816=6 then do; loanowed=750000; end;
    if YAS3816=7 then do; loanowed=1000001; end;
  end;
end;
if YAS3780=-1 or YAS3785=-1 or YAS3790=-1 or YAS3814L=-1 or YAS3814U=-1 or YAS3816=-1 then
loanowed=-1;
if YAS3780=-2 or YAS3785=-2 or YAS3790=-2 or YAS3814L=-2 or YAS3814U=-2 or YAS3816=-2 then
loanowed=-2;
if YAS3780=-3 or YAS3785=-3 or YAS3790=-3 or YAS3812=-3 or YAS3814L=-3 or YAS3814U=-3 or
YAS3816=-3 then loanowed=-3;

```

```

/* Remaining loans from round one still unpaid, total amount owed */
stilowed=0;
if (YAS3840=1 and YAS3850=1) or YAS3860=1 then do;
    if YAS3880=1 and YAS3882 ge 0 then stilowed=YAS3882;
    if YAS3880=2 and YAS3884U ge 0 and YAS3884L ge 0 then stilowed=(YAS3884L+(YAS3884U-YAS3884L)/2);
    if ((YAS3880 ne 1 and YAS3880 ne 2) or (YAS3882=-1 or YAS3882=-2)) then do;
        if YAS3886=1 then do; stilowed=500; end; if YAS3886=2 then do; stilowed=1750; end;
        if YAS3886=3 then do; stilowed=3750; end; if YAS3886=4 then do; stilowed=7500; end;
        if YAS3886=5 then do; stilowed=17500; end; if YAS3886=6 then do; stilowed=37500; end;
        if YAS3886=7 then do; stilowed=50001; end;
    end;
end;
if YAS3860=-1 or YAS3884L=-1 or YAS3884U=-1 or YAS3886=-1 then stilowed=-1;
if YAS3860=-2 or YAS3884L=-2 or YAS3884U=-2 or YAS3886=-2 then stilowed=-2;
if YAS3860=-3 or YAS3882=-3 or YAS3884L=-3 or YAS3884U=-3 or YAS3886=-3 then stilowed=-3;

/* Second mortgages */
secmortY=0;
if YAS3910=1 then do;
    if YAS3920=1 and YAS3922 ge 0 then secmortY=YAS3922;
    if YAS3920=2 and YAS3924U ge 0 and YAS3924L ge 0 then secmortY=(YAS3924L+(YAS3924U-YAS3924L)/2);
    if ((YAS3920 ne 1 and YAS3920 ne 2) or (YAS3922=-1 or YAS3922=-2)) then do;
        if YAS3926=1 then do; secmortY=500; end; if YAS3926=2 then do; secmortY=1750; end;
        if YAS3926=3 then do; secmortY=3750; end; if YAS3926=4 then do; secmortY=7500; end;
        if YAS3926=5 then do; secmortY=17500; end; if YAS3926=6 then do; secmortY=37500; end;
        if YAS3926=7 then do; secmortY=50001; end;
    end;
end;
if YAS3910=-1 or YAS3924L=-1 or YAS3924U=-1 or YAS3926=-1 then secmortY=-1;
if YAS3910=-2 or YAS3924L=-2 or YAS3924U=-2 or YAS3926=-2 then secmortY=-2;
if YAS3910=-3 or YAS3922=-3 or YAS3924L=-3 or YAS3924U=-3 or YAS3926=-3 then secmortY=-3;

/* Taxes on the property to be paid */
proptaxY=0;
if YAS3950=1 then do;
    if YAS3960=1 and YAS3962 ge 0 then proptaxY=YAS3962;
    if YAS3960=2 and YAS3964U ge 0 and YAS3964L ge 0 then proptaxY=(YAS3964L+(YAS3964U-YAS3964L)/2);
    if ((YAS3960 ne 1 and YAS3960 ne 2) or (YAS3962=-1 or YAS3962=-2)) then do;
        if YAS3966=1 then do; proptaxY=500; end; if YAS3966=2 then do; proptaxY=1750; end;
        if YAS3966=3 then do; proptaxY=3750; end; if YAS3966=4 then do; proptaxY=7500; end;
        if YAS3966=5 then do; proptaxY=17500; end; if YAS3966=6 then do; proptaxY=37500; end;
        if YAS3966=7 then do; proptaxY=50001; end;
    end;
end;
if YAS3950=-1 or YAS3964L=-1 or YAS3964U=-1 or YAS3966=-1 then proptaxY=-1;
if YAS3950=-2 or YAS3964L=-2 or YAS3964U=-2 or YAS3966=-2 then proptaxY=-2;
if YAS3950=-3 or YAS3962=-3 or YAS3964L=-3 or YAS3964U=-3 or YAS3966=-3 then proptaxY=-3;

/* Own a business, partnership or professional practice */
pvbuss=0;
if YAS4010=1 or YAS4010=2 then do;
    if YAS4030=1 then pvbuss=YAS4032;
    if YAS4030=2 then pvbuss=(YAS4034L+(YAS4034U-YAS4034L)/2);

```

```

if ((YAS4030 ne 1 and YAS4030 ne 2) or (YAS4032=-1 or YAS4032=-2)) then do;
    if YAS4036=1 then do; pvbuss=12500; end;
    if YAS4036=3 then do; pvbuss=75000; end;
    if YAS4036=5 then do; pvbuss=375000; end;
    if YAS4036=7 then do; pvbuss=1000001; end;
end;
end;

if YAS4010=-1 or YAS4034L=-1 or YAS4034U=-1 or YAS4036=-1 then pvbuss=-1;
if YAS4010=-2 or YAS4034L=-2 or YAS4034U=-2 or YAS4036=-2 then pvbuss=-2;
if YAS4010=-3 or YAS4032=-3 or YAS4034L=-3 or YAS4034U=-3 or YAS4036=-3 then pvbuss=-3;

/* Second real estate owned */
secrestY=0;
if YAS4140=1 or YAS4140=2 then do;
    if YAS4160=1 then secrestY=YAS4162;
    if YAS4160=2 then secrestY=(YAS4164L+(YAS4164U-YAS4164L)/2);
    if ((YAS4160 ne 1 and YAS4160 ne 2) or (YAS4162=-1 or YAS4162=-2)) then do;
        if YAS4166=1 then do; secrestY=12500; end;
        if YAS4166=3 then do; secrestY=75000; end;
        if YAS4166=5 then do; secrestY=375000; end;
        if YAS4166=7 then do; secrestY=1000001; end;
    end;
end;
if YAS4140=-1 or YAS4164L=-1 or YAS4164U=-1 or YAS4166=-1 then secrestY=-1;
if YAS4140=-2 or YAS4164L=-2 or YAS4164U=-2 or YAS4166=-2 then secrestY=-2;
if YAS4140=-3 or YAS4162=-3 or YAS4164L=-3 or YAS4164U=-3 or YAS4166=-3 then secrestY=-3;

/* Retirement plans or pensions */
retireY=0;
if YAS4270=1 or YAS4270=2 or YAS4270=3 then do;
    if YAS4290=1 and YAS4292 ge 0 then retireY=YAS4292;
    if YAS4290=2 and YAS4294U ge 0 and YAS4294L ge 0 then retireY=(YAS4294L+(YAS4294U-YAS4294L)/2);
    if ((YAS4290 ne 1 and YAS4290 ne 2) or (YAS4292=-1 or YAS4292=-2)) then do;
        if YAS4296=1 then do; retireY=2500; end;
        if YAS4296=3 then do; retireY=17500; end;
        if YAS4296=5 then do; retireY=75000; end;
        if YAS4296=7 then do; retireY=250001; end;
    end;
end;
if YAS4270=-1 or YAS4294L=-1 or YAS4294U=-1 or YAS4296=-1 then retireY=-1;
if YAS4270=-2 or YAS4294L=-2 or YAS4294U=-2 or YAS4296=-2 then retireY=-2;
if YAS4270=-3 or YAS4292=-3 or YAS4294L=-3 or YAS4294U=-3 or YAS4296=-3 then retireY=-3;

/* Savings in saving accounts, money market, funds, trusts,...*/
savingsY=0; flag4400=0;
if YAS4400=1 or YAS4400=2 or YAS4400=3 then do;
    if YAS4420=1 and YAS4422 ge 0 then savingsY=YAS4422;
    if YAS4420=2 and YAS4424U ge 0 and YAS4424L ge 0 then savingsY=(YAS4424L+(YAS4424U-YAS4424L)/2);
    if ((YAS4420 ne 1 and YAS4420 ne 2) or (YAS4422=-1 or YAS4422=-2)) then do;
        if YAS4426=1 then do; savingsY=500; flag4400=1; end;
        if YAS4426=2 then do; savingsY=1750; flag4400=1; end;
        if YAS4426=3 then do; savingsY=3750; flag4400=1; end;
        if YAS4426=4 then do; savingsY=7500; flag4400=1; end;
        if YAS4426=5 then do; savingsY=17500; flag4400=1; end;
        if YAS4426=6 then do; savingsY=37500; flag4400=1; end;
    end;
end;

```

```

        if YAS4426=7 then do; savingsY=50001; flag4400=1; end;
        end;
end;
if YAS4400=-1 or YAS4424L=-1 or YAS4424U=-2 or YAS4426=-1 then savingsY=-1;
if YAS4400=-2 or YAS4424L=-2 or YAS4424U=-2 or YAS4426=-2 then savingsY=-2;
if YAS4400=-3 or YAS4422=-3 or YAS4424L=-3 or YAS4424U=-3 or YAS4426=-3 then savingsY=-3;

/* Other savings in bonds or CDs */
othsavY=0;
if YAS4530=1 or YAS4530=2 or YAS4530=3 then do;
    if YAS4550=1 and YAS4552 ge 0 then othsavY=YAS4552;
    if YAS4550=2 and YAS4554U ge 0 and YAS4554L ge 0 then othsavY=(YAS4554L+(YAS4554U-
YAS4554L)/2);
    if ((YAS4550 ne 1 and YAS4550 ne 2) or (YAS4552=-1 or YAS4552=-2)) then do;
        if YAS4556=1 then do; othsavY=500; end; if YAS4556=2 then do; othsavY=1750; end;
        if YAS4556=3 then do; othsavY=3750; end; if YAS4556=4 then do; othsavY=7500; end;
        if YAS4556=5 then do; othsavY=17500; end; if YAS4556=6 then do; othsavY=37500; end;
        if YAS4556=7 then do; othsavY=50001; end;
    end;
end;
if YAS4530=-1 or YAS4554L=-1 or YAS4554U=-1 or YAS4556=-1 then othsavY=-1;
if YAS4530=-2 or YAS4554L=-2 or YAS4554U=-2 or YAS4556=-2 then othsavY=-2;
if YAS4530=-3 or YAS4552=-3 or YAS4554L=-3 or YAS4554U=-3 or YAS4556=-3 then othsavY=-3;

/* Stocks, mutual funds */
stockY=0;
if YAS4660=1 or YAS4660=2 or YAS4660=3 then do;
    if YAS4680=1 then stockY=YAS4682;
    if YAS4680=2 then stockY=(YAS4684L+(YAS4684U-YAS4684L)/2);
    if ((YAS4680 ne 1 and YAS4680 ne 2) or (YAS4682=-1 or YAS4682=-2)) then do;
        if YAS4686=1 then do; stockY=500; end; if YAS4686=2 then do; stockY=1750; end;
        if YAS4686=3 then do; stockY=3750; end; if YAS4686=4 then do; stockY=7500; end;
        if YAS4686=5 then do; stockY=17500; end; if YAS4686=6 then do; stockY=37500; end;
        if YAS4686=7 then do; stockY=50001; end;
    end;
end;
if YAS4660=-1 or YAS4684L=-1 or YAS4684U=-1 or YAS4686=-1 then stockY=-1;
if YAS4660=-2 or YAS4684L=-2 or YAS4684U=-2 or YAS4686=-2 then stockY=-2;
if YAS4660=-3 or YAS4682=-3 or YAS4684L=-3 or YAS4684U=-3 or YAS4686=-3 then stockY=-3;

/* Present value of vehicles owned */
pvcarsY=0; flag4790=0;
if YAS4790=1 or YAS4790=2 or YAS4790=3 then do;
    if YAS4810=1 then pvcarsY=YAS4812;
    if YAS4810=2 then pvcarsY=(YAS4814L+(YAS4814U-YAS4814L)/2);
    if ((YAS4810 ne 1 and YAS4810 ne 2) or (YAS4812=-1 or YAS4812=-2)) then do;
        if YAS4816=1 then do; pvcarsY=2500; flag4790=1; end;
        if YAS4816=2 then do; pvcarsY=7500; flag4790=1; end;
        if YAS4816=3 then do; pvcarsY=17500; flag4790=1; end;
        if YAS4816=4 then do; pvcarsY=37500; flag4790=1; end;
        if YAS4816=5 then do; pvcarsY=75000; flag4790=1; end;
        if YAS4816=6 then do; pvcarsY=175000; flag4790=1; end;
        if YAS4816=7 then do; pvcarsY=250001; flag4790=1; end;
    end;
end;
if YAS4790=-1 or YAS4814L=-1 or YAS4814U=-1 or YAS4816=-1 then pvcarsY=-1;

```

```

if YAS4790=-2 or YAS4814L=-2 or YAS4814U=-2 or YAS4816=-2 then pvcarsY=-2;
if YAS4790=-3 or YAS4812=-3 or YAS4814L=-3 or YAS4814U=-3 or YAS4376=-3 then pvcarsY=-3;

/* Money still owed on these vehicles */
cardebt=0; flag4840=0;
if YAS4840=1 then cardebt=YAS4842;
if YAS4840=2 then cardebt=(YAS4844L+(YAS4844U-YAS4844L)/2);
if ((YAS4840 ne 1 and YAS4840 ne 2) or (YAS4842=-1 or YAS4842=-2)) then do;
    if YAS4846=1 then do; cardebt=2500; flag4840=1; end;
    if YAS4846=2 then do; cardebt=7500; flag4840=1; end;
    if YAS4846=3 then do; cardebt=17500; flag4840=1; end;
    if YAS4846=4 then do; cardebt=37500; flag4840=1; end;
    if YAS4846=5 then do; cardebt=75000; flag4840=1; end;
    if YAS4846=6 then do; cardebt=175000; flag4840=1; end;
    if YAS4846=7 then do; cardebt=250001; flag4840=1; end;
end;
if YAS4790=-1 or YAS4844L=-1 or YAS4844U=-1 or YAS4846=-1 then cardebt=-1;
if YAS4790=-2 or YAS4844L=-2 or YAS4844U=-2 or YAS4846=-2 then cardebt=-2;
if YAS4790=-3 or YAS4842=-3 or YAS4844L=-3 or YAS4844U=-3 or YAS4376=-3 then cardebt=-3;

/* Present value of owned furniture */
pvfurnty=0;
if YAS4870=1 then do; pvfurnty=2500; end;           if YAS4870=2 then do; pvfurnty=7500; end;
if YAS4870=3 then do; pvfurnty=17500; end;         if YAS4870=4 then do; pvfurnty=37500; end;
if YAS4870=5 then do; pvfurnty=75000; end;         if YAS4870=6 then do; pvfurnty=175000; end;
if YAS4870=7 then do; pvfurnty=250001; end;
if YAS4870=-1 then pvfurnty=-1;
if YAS4870=-2 then pvfurnty=-2;
if YAS4870=-3 then pvfurnty=-3;

/* Other assets not being mentioned before */
otassetY=0;
if YAS4880=1 or YAS4880=2 or YAS4880=3 then do;
    if YAS4900=1 then otassetY=YAS4902;
    if YAS4900=2 then otassetY=(YAS4904L+(YAS4904U-YAS4904L)/2);
    if ((YAS4900 ne 1 and YAS4900 ne 2) or (YAS4902=-1 or YAS4902=-2)) then do;
        if YAS4906=1 then do; otassetY=2500; end;   if YAS4906=2 then do; otassetY=7500; end;
        if YAS4906=3 then do; otassetY=17500; end;  if YAS4906=4 then do; otassetY=37500; end;
        if YAS4906=5 then do; otassetY=75000; end;  if YAS4906=6 then do; otassetY=175000; end;
        if YAS4906=7 then do; otassetY=250001; end;
    end;
end;
if YAS4880=-1 or YAS4904L=-1 or YAS4904U=-1 or YAS4906=-1 then otassetY=-1;
if YAS4880=-2 or YAS4904L=-2 or YAS4904U=-2 or YAS4906=-2 then otassetY=-2;
if YAS4880=-3 or YAS4902=-3 or YAS4904L=-3 or YAS4904U=-3 or YAS4376=-3 then otassetY=-3;

/* Loans still owed to family or relatives */
array rloan rloan01 rloan02 rloan03 rloan04 rloan05 rloan06 rloan07 rloan08 rloan09 rloan10 rloan11 rloan12
rloan13 rloan14 rloan15 rloan16 rloan17 rloan18 rloan19 rloan20;
array A5080 A508001 A508002 A508003 A508004 A508005 A508006 A508007 A508008 A508009 A508010
A508011 A508012 A508013 A508014 A508015 A508016 A508017 A508018 A508019 A508020;
array A5082 A508201 A508202 A508203 A508204 A508205 A508206 A508207 A508208 A508209 A508210
A508211 A508212 A508213 A508214 A508215 A508216 A508217 A508218 A508219 A508220;
array A5084L A5084L01 A5084L02 A5084L03 A5084L04 A5084L05 A5084L06 A5084L07 A5084L08 A5084L09
A5084L10 A5084L11 A5084L12 A5084L13 A5084L14 A5084L15 A5084L16 A5084L17 A5084L18
A5084L19 A5084L20;

```

```

array A5084U A5084U01 A5084U02 A5084U03 A5084U04 A5084U05 A5084U06 A5084U07 A5084U08
      A5084U09 A5084U10 A5084U11 A5084U12 A5084U13 A5084U14 A5084U15 A5084U16 A5084U17
      A5084U18 A5084U19 A5084U20;
array A5086 A508601 A508602 A508603 A508604 A508605 A508606 A508607 A508608 A508609 A508610
      A508611 A508612 A508613 A508614 A508615 A508616 A508617 A508618 A508619 A508620;
array A5170 A517001 A517002 A517003 A517004 A517005 A517006 A517007 A517008 A517009 A517010
      A517011 A517012 A517013 A517014 A517015 A517016 A517017 A517018 A517019 A517020;
array A5172 A517201 A517202 A517203 A517204 A517205 A517206 A517207 A517208 A517209 A517210
      A517211 A517212 A517213 A517214 A517215 A517216 A517217 A517218 A517219 A517220;
array A5174L A5174L01 A5174L02 A5174L03 A5174L04 A5174L05 A5174L06 A5174L07 A5174L08 A5174L09
      A5174L10 A5174L11 A5174L12 A5174L13 A5174L14 A5174L15 A5174L16 A5174L17 A5174L18
      A5174L19 A5174L20;
array A5174U A5174U01 A5174U02 A5174U03 A5174U04 A5174U05 A5174U06 A5174U07 A5174U08
      A5174U09 A5174U10 A5174U11 A5174U12 A5174U13 A5174U14 A5174U15 A5174U16 A5174U17
      A5174U18 A5174U19 A5174U20;
array A5176 A517601 A517602 A517603 A517604 A517605 A517606 A517607 A517608 A517609 A517610
      A517611 A517612 A517613 A517614 A517615 A517616 A517617 A517618 A517619 A517620;
array A5074 A507401-A507420;
array A5066 A506601-A506620;
array A5164 A516401-A516420;

do I=1 to 20;
rloan(I)=0;
if YAS5040=1 and YAS5060 ne 1 and YAS5062 ne 1 then do;
  if A5080(I)=1 then rloan(I)=A5082(I);
  if A5080(I)=2 then rloan(I)=(A5084L(I)+(A5084U(I)-A5084L(I))/2);
  if ((A5080(I) ne 1 and A5080(I) ne 2) or (A5082(I)=-1 or A5082(I)=-2)) then do;
    if A5086(I)=1 then do; rloan(I)=500; end;           if A5086(I)=2 then do; rloan(I)=1750; end;
    if A5086(I)=3 then do; rloan(I)=3750; end;         if A5086(I)=4 then do; rloan(I)=7500; end;
    if A5086(I)=5 then do; rloan(I)=17500; end;        if A5086(I)=6 then do; rloan(I)=37500; end;
    if A5086(I)=7 then do; rloan(I)=50001; end;
  end;
end;
if YAS5040=-1 or A5074(I)=-1 or A5086(I)=-1 or A5084L[I]=-1 or A5084U[I]=-1 then rloan(I)=-1;
if YAS5040=-2 or A5074(I)=-2 or A5086(I)=-2 or A5084L[I]=-2 or A5084U[I]=-2 then rloan(I)=-2;
if YAS5040=-3 or A5082(I)=-3 or A5086(I)=-3 or A5084L[I]=-3 or A5084U[I]=-3 then rloan(I)=-3;
if YAS5130=1 and YAS5150 ne 1 and YAS5152 ne 1 then do;
  if A5170(I)=1 then rloan(I)=A5172(I);
  if A5170(I)=2 then rloan(I)=(A5174L(I)+(A5174U(I)-A5174L(I))/2);
  if ((A5170(I) ne 1 and A5170(I) ne 2) or (A5172(I)=-1 or A5172(I)=-2)) then do;
    if A5176(I)=1 then do; rloan(I)=500; end;           if A5176(I)=2 then do; rloan(I)=1750; end;
    if A5176(I)=3 then do; rloan(I)=3750; end;         if A5176(I)=4 then do; rloan(I)=7500; end;
    if A5176(I)=5 then do; rloan(I)=17500; end;        if A5176(I)=6 then do; rloan(I)=37500; end;
    if A5176(I)=7 then do; rloan(I)=50001; end;
  end;
end;
if YAS5130=-1 or A5164(I)=-1 or A5176(I)=-1 or A5174L[I]=-1 or A5174U[I]=-1 then rloan(I)=-1;
if YAS5130=-2 or A5164(I)=-2 or A5176(I)=-2 or A5174L[I]=-2 or A5174U[I]=-2 then rloan(I)=-2;
if YAS5130=-3 or A5172(I)=-3 or A5176(I)=-3 or A5174L[I]=-3 or A5174U[I]=-3 then rloan(I)=-3;
end;

/* Other debts from loans, credit cards, etc...*/
othdebtY=0;
if YAS5210=1 then do;
  if YAS5220=1 then othdebtY=YAS5222;
  if YAS5220=2 then othdebtY=(YAS5224L+(YAS5224U-YAS5224L)/2);

```

```

if ((YAS5220 ne 1 and YAS5220 ne 2) or (YAS5222=-1 or YAS5222=-2)) then do;
  if YAS5226=1 then do; othdebtY=500; end;
  if YAS5226=3 then do; othdebtY=3750; end;
  if YAS5226=5 then do; othdebtY=17500; end;
  if YAS5226=7 then do; othdebtY=50001; end;
end;
end;

if YAS5210=-1 or YAS5224L=-1 or YAS5224U=-1 or YAS5226=-1 then othdebtY=-1;
if YAS5210=-2 or YAS5224L=-2 or YAS5224U=-2 or YAS5226=-2 then othdebtY=-2;
if YAS5210=-3 or YAS5222=-3 or YAS5224L=-3 or YAS5224U=-3 or YAS5226=-3 then othdebtY=-3;

/* We now calculate the household net worth according to the youth: hhworthY=assets-liabilities*/
if YAS50=1 then do;
  hhworthY=0;
  if pvranch not in (-1, -2, -3) then hhworthY=hhworthY+pvranch;
  if pvmbst not in (-1, -2, -3) then hhworthY=hhworthY+pvmbst;
  if pvmb not in (-1, -2, -3) then hhworthY=hhworthY+pvmb;
  if pvst not in (-1, -2, -3) then hhworthY=hhworthY+pvst;
  if pvapthm not in (-1, -2, -3) then hhworthY=hhworthY+pvapthm;
  if pbuss not in (-1, -2, -3) then hhworthY=hhworthY+pbuss;
  if secrestY not in (-1, -2, -3) then hhworthY=hhworthY+secrestY;
  if retireY not in (-1, -2, -3) then hhworthY=hhworthY+retireY;
  if savingsY not in (-1, -2, -3) then hhworthY=hhworthY+savingsY;
  if othsavY not in (-1, -2, -3) then hhworthY=hhworthY+othsavY;
  if stockY not in (-1, -2, -3) then hhworthY=hhworthY+stockY;
  if pvcarsY not in (-1, -2, -3) then hhworthY=hhworthY+pvcarsY;
  if pfurnty not in (-1, -2, -3) then hhworthY=hhworthY+pfurnty;
  if otassetY not in (-1, -2, -3) then hhworthY=hhworthY+otassetY;
  if mortgagY not in (-1, -2, -3) then hhworthY=hhworthY-mortgagY;
  if loanowed not in (-1, -2, -3) then hhworthY=hhworthY-loanowed;
  if stilowed not in (-1, -2, -3) then hhworthY=hhworthY-stilowed;
  if secmortY not in (-1, -2, -3) then hhworthY=hhworthY-secmortY;
  if cardebt not in (-1, -2, -3) then hhworthY=hhworthY-cardebt;
  if othdebtY not in (-1, -2, -3) then hhworthY=hhworthY-othdebtY;
  do I=1 to 20; if rloan[I] not in (-1, -2, -3) then hhworthY=hhworthY-rloan[I]; end;
end;

if YAS50=0 or YAS50=-4 then hhworthY=-4;

if YAS50=-1 or (pvranch=-1 or pvmbast=-1 or pvmbhm=-1 or pvmbst=-1 or pvapthm=-1 or pbuss=-1 or secrestY=-1
  or retireY=-1 or savingsY=-1 or othsavY=-1 or stockY=-1 or pvcarsY=-1 or pfurnty=-1 or otassetY=-1
  or mortgagY=-1 or loanowed=-1 or stilowed=-1 or secmortY=-1 or cardebt=-1 or rloan01=-1 or rloan02=-1
  or rloan03=-1 or rloan04=-1 or rloan05=-1 or rloan06=-1 or rloan07=-1 or rloan08=-1 or rloan09=-1 or
  rloan10=-1 or rloan11=-1 or rloan12=-1 or rloan13=-1 or rloan14=-1 or rloan15=-1 or rloan16=-1 or
  rloan17=-1 or rloan18=-1 or rloan19=-1 or rloan20=-1 or othdebtY=-1) then hhworthY=-1;
if YAS50=-2 or (pvranch=-2 or pvmbast=-2 or pvmbhm=-2 or pvmbst=-2 or pvapthm=-2 or pbuss=-2 or secrestY=-2
  or retireY=-2 or savingsY=-2 or othsavY=-2 or stockY=-2 or pvcarsY=-2 or pfurnty=-2 or otassetY=-2
  or mortgagY=-2 or loanowed=-2 or stilowed=-2 or secmortY=-2 or cardebt=-2 or rloan01=-2 or rloan02=-2
  or rloan03=-2 or rloan04=-2 or rloan05=-2 or rloan06=-2 or rloan07=-2 or rloan08=-2 or rloan09=-2 or
  rloan10=-2 or rloan11=-2 or rloan12=-2 or rloan13=-2 or rloan14=-2 or rloan15=-2 or rloan16=-2 or
  rloan17=-2 or rloan18=-2 or rloan19=-2 or rloan20=-2 or othdebtY=-2) then hhworthY=-2;
if YAS50=-3 or (pvranch=-3 or pvmbast=-3 or pvmbhm=-3 or pvmbst=-3 or pvapthm=-3 or pbuss=-3 or secrestY=-3
  or retireY=-3 or savingsY=-3 or othsavY=-3 or stockY=-3 or pvcarsY=-3 or pfurnty=-3 or otassetY=-3
  or mortgagY=-3 or loanowed=-3 or stilowed=-3 or secmortY=-3 or cardebt=-3 or rloan01=-3 or rloan02=-3
  or rloan03=-3 or rloan04=-3 or rloan05=-3 or rloan06=-3 or rloan07=-3 or rloan08=-3 or rloan09=-3 or
  rloan10=-3 or rloan11=-3 or rloan12=-3 or rloan13=-3 or rloan14=-3 or rloan15=-3 or rloan16=-3 or
  rloan17=-3 or rloan18=-3 or rloan19=-3 or rloan20=-3 or othdebtY=-3) then hhworthY=-3;

```

```
rloan10=-3 or rloan11=-3 or rloan12=-3 or rloan13=-3 or rloan14=-3 or rloan15=-3 or rloan16=-3 or
rloan17=-3 or rloan18=-3 or rloan19=-3 or rloan20=-3 or othdebtY=-3) then hhworthY=-3;
if YAS50=-5 then hhworthY=-5; /*give the people who were not interviewed in round 3 value -5.*/
```

```
*****hand edit for round 3****/ if pubid=429 then hhworthy=-3;
```

```
hhworthY=round(hhworthY,1);
```

******* SECTION 3: HOUSEHOLD POVERTY STATUS *******

```
povert=-4;
```

```
if HHSIZE=-1 or under18=-1 then povert=-1; if HHSIZE=-2 or under18=-2 then povert=-2;
if HHSIZE=-3 or under18=-3 then povert=-3;
```

```
if HHSIZE=1 then povert=8480;
```

```
if HHSIZE=2 then do;
```

```
    if under18=0 then povert=10915;  if under18=1 then povert=11235;  if under18=2 then povert=11235;
end;
```

```
if HHSIZE=3 then do;
```

```
    if under18=0 then povert=12750;  if under18=1 then povert=13120;
    if under18=2 then povert=13133;  if under18=3 then povert=13133;
end;
```

```
if HHSIZE=4 then do;
```

```
    if under18=0 then povert=16813;  if under18=1 then povert=17088;  if under18=2 then povert=16530;
    if under18=3 then povert=16588;  if under18=4 then povert=16588;
end;
```

```
if HHSIZE=5 then do;
```

```
    if under18=0 then povert=20275;  if under18=1 then povert=20570;  if under18=2 then povert=19940;
    if under18=3 then povert=19453;  if under18=4 then povert=19155;  if under18=5 then povert=19155;
end;
```

```
if HHSIZE=6 then do;
```

```
    if under18=0 then povert=23320;  if under18=1 then povert=23413;  if under18=2 then povert=22930;
    if under18=3 then povert=22468;  if under18=4 then povert=21780;  if under18=5 then povert=21373;
    if under18=6 then povert=21373;
end;
```

```
if HHSIZE=7 then do;
```

```
    if under18=0 then povert=26833;  if under18=1 then povert=27000;  if under18=2 then povert=26423;
    if under18=3 then povert=26020;  if under18=4 then povert=25270;  if under18=5 then povert=24395;
    if under18=6 then povert=23435;  if under18=7 then povert=23435;
end;
```

```
if HHSIZE=8 then do;
```

```
    if under18=0 then povert=30010;  if under18=1 then povert=30275;  if under18=2 then povert=29730;
    if under18=3 then povert=29253;  if under18=4 then povert=28575;  if under18=5 then povert=27715;
    if under18=6 then povert=26820;  if under18=7 then povert=26593;  if under18=8 then povert=26593;
end;
```

```
if HHSIZE>=9 then do;
```

Appendix 5: Income and Assets Variable Creation

```

if under18=0 then povert=36100; if under18=1 then povert=36275; if under18=2 then povert=35793;
if under18=3 then povert=35388; if under18=4 then povert=34723; if under18=5 then povert=33808;
if under18=6 then povert=32980; if under18=7 then povert=32775; if under18=8 then povert=31513;
if under18=9 then povert=31513; if under18=10 then povert=31513; if under18=11 then povert=31513;
end;

povthr=-4;
if groshhIY ge 0 and povert gt 0 then povthr=groshhIY/povert;
if groshhIY eq -1 or povert=-1 then povthr=-1; if groshhIY eq -2 or povert=-2 then povthr=-2;
if groshhIY eq -3 or povert=-3 then povthr=-3; if groshhIY eq -4 then povthr=-4;

povthr2=-4;
if povthr ge 0 then do; povthr2=povthr*100; end;
if povthr eq -1 then povthr2=-1; if povthr eq -2 then povthr2=-2; if povthr eq -3 then povthr2=-3;

povthr3=-4;
if povthr2 ge 0 then do; povthr3=round(povthr2, 1); end;
if povthr2 eq -1 then povthr3=-1; if povthr2 eq -2 then povthr3=-2; if povthr2 eq -3 then povthr3=-3;

/* Respondents who were not interviewed in round 3 */
if YAS50=-5 then povthr3=-5;

```

***** SECTION 4: TOPCODING INCOME, NET WORTH, AND POVERTY RATIO *****

```

/* If hhincome, the poverty ratio, or the household net worth exceed the lowest value of the top 2% cases, then they
are topcoded by the mean value corresponding to the top 2% cases */
if hhincome ge 212500 then hhincome=291681;
if povthr3 ge 1269 then povthr3=2009;
if hhworthy ge 213900 then hhworthy=619834;

```

endsas;

PARTICIPATION IN GOVERNMENT PROGRAMS

Variables Created: CV_AMT_GOVNT_PGM_PCY.80 – CV_AMT_GOVNT_PGM_PCY.00
CV_GOVNT_PGM_EVER
CV_GOVNT_PGM_YR.80 – CV_GOVNT_PGM_YR.00

Variables Used

| Name in Program | Question Name on CD | Name in Program | Question Name on CD |
|--------------------|-------------------------|--------------------------------------|---|
| int97_d,_m,_y | YINF-900_D,_M,_Y | ps1810-1820, ps18301, ps1840-1930 | YPRG-1810_UPD-YPRG-1930_UPD |
| int98_d,_m,_y | YINTDATE-D,-M,-Y | ps40001m, y-ps40002m, y | YPRG-4000_UPD.01-M, ~Y-.02-M, ~Y |
| p010 | YPRG-010 | ps41001-ps45001, ps41002-ps45002 | YPRG-4100_UPD.01-YPRG-4500_UPD.01,.02 |
| p1210 | YPRG-1210 | ps48001-ps61001, ps54002-ps61002 | YPRG-4800_UPD.01-YPRG-6100_UPD.01,.02 |
| p1400 | YPRG-1400 | ps9780-9790, ps98001, ps9810-9900 | YPRG-9780_UPD-YPRG-9900_UPD |
| ps390101 | YPRG-3901_UPD.01 | ps103001-ps106001 | YPRG-10300_UPD.01-YPRG-10600_UPD.01 |
| ps47001m, y | YPRG-4700_UPD.01-M, ~Y | | |
| ps6200b1, ps6200b2 | YPRG-6200B_UPD.01,.02 | | |
| ps63001 | YPRG-6300_UPD.01 | | |
| ps10100a1 | YPRG-10100A_UPD.01 | ps110001-ps124001 | YPRG-11000_UPD.01-YPRG-12400_UPD.01 |
| ps102001m, y | YPRG-10200_UPD.01-M, ~Y | ps16290-ps16385 | YPRG-16290_UPD-YPRG-16385_UPD |
| ps109001m, y | YPRG-10900_UPD.01-M, ~Y | ps16400, ps16410 | YPRG-16400_UPD, YPRG-16410_UPD |
| ps125001 | YPRG-12500A_UPD.01 | ps191001m, y-ps191002m, y | YPRG-19100_UPD.01-M, ~Y-.02-M, ~Y |
| ps12600a1 | YPRG-12600_UPD.01 | ps200001m, y-ps200002m, y | YPRG-20000_UPD.01-M, ~Y-.02-M, ~Y |
| ps16390m, y | YPRG-16390_UPD-M, ~Y | ps201001-ps208001, ps201002-ps207002 | YPRG-20100_UPD.01-YPRG-20800_UPD.01,.02 |
| ps19063a1 | YPRG-19063A_UPD.01 | | |
| ps191201 ps191202 | YPRG-19120_UPD.01,.02 | | |

Appendix 5: Income and Assets Variable Creation

| | | | |
|---------------------------|-------------------------|-----------------------------------|---------------------------------------|
| ps191401 ps191402 | YPRG-19140_UPD.01, .02 | ps2090011-ps2090015 | YPRG-20900_UPD.01~000001~~000005 |
| ps197001 ps197002 | YPRG-19700_UPD.01, .02 | ps2090021-ps2090025 | YPRG-20900_UPD.02~000001~~000005 |
| ps210001, ps210002 | YPRG-21000_UPD.01, .02 | ps21310-ps21405 | YPRG-21310_UPD-YPRG-21405_UPD |
| ps21100a1, ps21100a2 | YPRG-21100A_UPD.01, .02 | ps21420, ps21430 | YPRG-21420_UPD, YPRG-21430_UPD |
| ps212001, ps212002 | YPRG-21200_UPD.01, .02 | ps220001m, y-ps220002m, y | YPRG-22000_UPD.01-M, ~Y.02-M, ~Y |
| ps21410m, y | YPRG-21410_UPD-M, ~Y | ps221001-ps223001, ps221002- | YPRG-22100_UPD.01-YPRG- |
| ps21900a1 | YPRG-21900A_UPD.01 | ps223002 | 22300_UPD.01, .02 |
| ps236001, ps236002 | YPRG-23600_UPD.01, .02 | ps226001m, y-ps226002m, y | YPRG-22600_UPD.01-M, ~Y.02-M, ~Y |
| ps23700a1, ps23700a2 | YPRG-23700A_UPD.01, .02 | ps227001-ps234001, ps227002- | YPRG-22700_UPD.01-YPRG- |
| ps238001 | YPRG-23800_UPD.01 | ps234002 | 23400_UPD.01, .02 |
| ps35720m, y | YPRG-35720_UPD-M, ~Y | ps2350011-ps2350015 | YPRG-23500_UPD.01-000001~~000005 |
| ps357901 | YPRG-35790_UPD.01 | ps2350021-ps2350025 | YPRG-23500_UPD.02-000001~~000005 |
| ps358401m, y | YPRG-35840_UPD.01-M, ~Y | ps35610-ps35715 | YPRG-35610_UPD-YPRG-35715_UPD |
| ps359301 ps359302 | YPRG-35930_UPD.01, .02 | ps35730, ps35740 | YPRG-35730_UPD, YPRG-35740_UPD |
| ps359501 ps359502 | YPRG-35950_UPD.01, .02 | ps358001m, y-ps358002m, y | YPRG-35800_UPD.01-M, ~Y.02-M, ~Y |
| ps359601 | YPRG-35960_UPD.01 | ps358101-ps358301, ps358102- | YPRG-35810_UPD.01-YPRG- |
| ps36100m, y | YPRG-36100_UPD-M, ~Y | ps358302 | 35830_UPD.01, .02 |
| ps30900a1 | YPRG-30900A_UPD.01 | ps358501-ps359101, ps358502- | YPRG-35850_UPD.01-YPRG- |
| ps310001m, y | YPRG-31000_UPD.01-M, ~Y | ps359002 | 35910_UPD.01, .02 |
| ps310002m, y | YPRG-31000_UPD.02-M, ~Y | ps3592011 ps3592015 | YPRG-35920_UPD.01-000001~~000005 |
| ps316001m, y | YPRG-31600_UPD.01-M, ~Y | ps3592021 ps3592025 | YPRG-35920_UPD.02-000001~~000005 |
| ps326001, ps326002 | YPRG-32600_UPD.01, .02 | ps36000-ps36095 | YPRG-36000_UPD-YPRG-36095_UPD |
| ps32700a1, ps32700a2 | YPRG-32700A_UPD.01, .02 | ps36110, ps36120 | YPRG-36110_UPD, YPRG-36120_UPD |
| ps329001 | YPRG-32900_UPD.01 | ps311001-ps313001, ps311002- | YPRG-31100_UPD.01-YPRG- |
| p1900, p2000 | YPRG-1900, YPRG-2000 | ps313002 | 31300_UPD.01, .02 |
| p2500-p3500 | YPRG-2500-YPRG-3500 | ps317001-ps324001, ps317002- | YPRG-31700_UPD.01-YPRG- |
| p40001m, y | YPRG-4000.01-M, ~Y | ps324002 | 32400_UPD.01, .02 |
| p47001m, y | YPRG-4700.01-M, ~Y | ps3250011-ps3250015 | YPRG-32500_UPD.01-000001~~000005 |
| p9550ec | YPRG-9550-ELIG-CHECK | ps3250021-ps3250025 | YPRG-32500_UPD.02-000001~~000005 |
| p9600, p9700 | YPRG-9600, YPRG-9700 | p41001-p45901 | YPRG-4100.01-YPRG-4590.01 |
| p16010ec | YPRG-16010-ELIG-CHECK | p48001-p63001 | YPRG-4800.01-YPRG-6300.01 |
| p18800, p18900 | YPRG-18800, YPRG-18900 | p102001m, y-p102003m, y | YPRG-10200.01-M, ~Y.03-M, ~Y |
| p200001m, y | YPRG-20000.01-M, ~Y | p103001-p107901, p103002-p107902, | YPRG-10300.01-YPRG-10790.01, .02, .03 |
| p210001, p210002 | YPRG-21000.01, .02 | p103003-p107903 | |
| p212001 | YPRG-21200.01 | p109001m, y-p109003m, y | YPRG-10900.01-M, ~Y.03-M, ~Y |
| p21300ec | YPRG-21300-ELIG-CHECK | p110001-p126001, p110002-p126002, | YPRG-11000.01-YPRG-12600.01, .02, .03 |
| p21400, p21500 | YPRG-21400, YPRG-21500 | p110003-p124003 | |
| p226001m, y | YPRG-22600.01-M, ~Y | p194001m, y-p194002m, y | YPRG-19400.01-M, ~Y.02-M, ~Y |
| p236001, p236002 | YPRG-23600.01, .02 | p195001-p197901, p195002-p197902 | YPRG-19500.01-YPRG-19790.01, .02 |
| p238001 | YPRG-23800.01 | p201001-p208001, p201002-p207002 | YPRG-20100.01-YPRG-20800.01, .02 |
| p23900ec | YPRG-23900-ELIG-CHECK | p2090011-p2090015 | YPRG-20900.01-000001~~000005 |
| p35600ec | YPRG-35600-ELIG-CHECK | p2090021-p2090025 | YPRG-20900.02-000001~~000005 |
| p16100, p16200 | YPRG-16100, YPRG-16200 | p220001m, y-p220002m, y | YPRG-22000.01-M, ~Y.02-M, ~Y |
| p172001m, y | YPRG-17200.01-M, ~Y | p221001-p223901, p221002-p223902 | YPRG-22100.01-YPRG-22390.01, .02 |
| p184001, p184002 | YPRG-18400.01, .02 | p227001-p234001, p230002-p233002 | YPRG-22700.01-YPRG-23400.01, .02 |
| p186001 | YPRG-18600.01 | p2350011-p2350015 | YPRG-23500.01-000001~~000005 |
| p18700ec | YPRG-18700-ELIG-CHECK | p2350021-p2350025 | YPRG-23500.02-000001~~000005 |
| p29100, p30500 | YPRG-29100, YPRG-30500 | p167001m, y-p167002m, y | YPRG-16700.01-M, ~Y.02-M, ~Y |
| p326001, p326002 | YPRG-32600.01, .02 | p168001-p170901, p168002-p170902 | YPRG-16800.01-YPRG-17090.01, .02 |
| p329001, p329002 | YPRG-32900.01, .02 | p173001-p181001, p177002-p181002 | YPRG-17300.01-YPRG-18100.01, .02 |
| ui1total80-ui1total98 | CV_UI_WKS_YR.80-.98 | p1830011-p1830015 | YPRG-18300.01-000001~~000005 |
| ui1 Ever | CV_UI_WKS_EVER | p18300210-p1830025 | YPRG-18300.02-000001~~000005 |
| ui1_spells80-ui1_spells98 | CV_UI_SPELLS_YR.80-.98 | p310001m, y-p310002m, y | YPRG-31000.01-M, ~Y.02-M, ~Y |
| ui1total80-ui1total98 | CV_AMT_UI_YR.80-.98 | p311001-p313901, p311002-p313902 | YPRG-31100.01-YPRG-31390.01, .02 |
| ui2total80-ui2total99 | CV_UI_YR.80-.99 | p316001m, y-p316002m, y | YPRG-31600.01-M, ~Y.02-M, ~Y |
| ui2 Ever | CV_UI_EVER | p317001-p324001, p317002-p323002 | YPRG-31700.01-YPRG-32400.01, .02 |
| ui2_spells80-ui2_spells99 | CV_UI_SPELLS_YR.80-.99 | p3250011-p3250015 | YPRG-32500.01-000001~~000005 |
| ui2total97-ui2total99 | CV_AMT_UI_YR.97-.99 | p3250021-p3250025 | YPRG-32500.02-000001~~000005 |
| wc1total80-wc1total98 | CV_WC_WKS_YR.80-.98 | bdate_d, _m, _y | SYMBOL!KEY!BDATE-D, ~M, ~Y |
| wc1 Ever | CV_WC_WKS | yint_d, _m, _y | SYMBOL!CURDATE-D, ~M, ~Y |
| wca1total80-wca1total98 | CV_AMT_WC_YR.80-.98 | g1total80-g1total98 | CV_GOVNT_PRG_YR.80-.98 |
| wc2total80-wc2total99 | CV_WC_YR.80-.99 | g1prg_ever | CV_GOVNT_PRG_EVER |
| wc2 Ever | CV_WC_EVER | ga1total80-ga1total98 | CV_AMT_GOVNT_PGM_PCY.80-.98 |
| wca2total81-wca2total99 | CV_AMT_WC_YR.81-.99 | g2total80-g2total99 | CV_GOVNT_PRG_YR.80-.99 |
| pubid | PUBID | g2prg_ever | CV_GOVNT_PRG_EVER |
| | | ga2total80-ga2total99 | CV_AMT_GOVNT_PGM_PCY.80-.99 |

This program creates several variables describing the respondent's participation in government programs for the economically disadvantaged. During the interview, respondents report amounts received and months of participation in Aid to Families with Dependent Children (AFDC); food stamps; and Women, Infants, and Children (WIC). There is also an "other assistance" question to capture information about any other government program from which respondents may have received assistance. The interview also records amounts received from unemployment compensation and worker's compensation in separate question series; this program also creates variables describing respondents' participation in unemployment or worker's compensation.

The program to create these variables first creates a month-by-month participation array for each of the six categories (AFDC, food stamps, WIC, other programs, unemployment compensation, and worker's compensation). These month-by-month variables constitute part of the event history array for program participation; see appendix 7 for more information. After all six arrays are created, the program merges data from the six categories to create the summary variables.

```

*****hand edits - these respondents reported data that are inconsistent*****
if pubid=3921 then do;          /*reported ui stop before ui start*/
    PS40001M=-4; PS40001Y=-4; PS47001M=-4; PS47001Y=-4; end;
if pubid=4856 then do;          /*reported ui for 1 day and zero for amount*/
    P40001M=-4; P40001Y=-4; P47001M=-4; P47001Y=-4; end;
if pubid=8348 then do;          /*reported wc in 1983/1990 when never employed & below age*/
    P102001M=-4; P102001Y=-4; P109001M=-4; P109001Y=-4; P102002M=-4; P102002Y=-4; P109002M=-4;
    P109002Y=-4; P102003M=-4; P102003Y=-4; P109003M=-4; P109003Y=-4; P114003=-4; P116002=-4;
    P116003=-4; P117002=-4; P117003=-4; P118002=-4; P118003=-4; P121002=-4; P121003=-4; P123002=-4;
    P123003=-4; end;
if pubid=1364 then do;          /*reported wc for 1 day and zero for amount*/
    PS102001M=-4; PS102001Y=-4; PS109001M=-4; PS109001Y=-4; end;
if pubid=4812 then do;          /*reported wic, but never involved with birth of child*/
    P220001M=-4; P220001Y=-4; P226001M=-4; P226001Y=-4; end;
if pubid=6994 then do;          /*reported wic, but never involved with birth of child*/
    P220001M=-4; P220001Y=-4; P226001M=-4; P226001Y=-4; end;
if pubid=2908 then do;          /*correct for a 0 in where year should be*/
    P470011Y=-2; end;

*change weekly ui figures to daily amount;
if PS59001 gt 0 then ps59001=ps59001/7; if PS59002 gt 0 then ps59002=ps59002/7;
if PS59003 gt 0 then ps59003=ps59003/7; if P59001 gt 0 then p59001=p59001/7;
if P59002 gt 0 then p59002=p59002/7; if P59003 gt 0 then p59003=p59003/7;

*change weekly wc figures to daily amount;
if PS121001 gt 0 then ps121001=ps121001/7; if PS121002 gt 0 then ps121002=ps121002/7;
if PS121003 gt 0 then ps121003=ps121003/7; if P121001 gt 0 then p121001=p121001/7;
if P121002 gt 0 then p121002=p121002/7; if P121003 gt 0 then p121003=p121003/7;

/*change previous round variables to account for noninterviews*/
array gtotal (i) gtotal80-gtotal99 gtotal00;           array g2total (i) g2total80-g2total99 g2total00;
array g1total (i) g1total80-g1total99 g1total00;         array gatotal (i) gatotal80-gatotal99 gatotal00;
array ga2total (i) ga2total80-ga2total99 ga2total00;       array ga1total (i) ga1total80-ga1total99 ga1total00;
array uitotal (i) uitotal80-uitotal99 uitotal00;         array ui2total (i) ui2total80-ui2total99 ui2total00;
array ui1total (i) ui1total80-ui1total99 ui1total00;       array uiatotal (i) uiatotal80-uiatotal99 uiatotal00;
array uia2total (i) uia2total80-uaia2total99 uia2total00;   array uia1total (i) uia1total80-uaia1total99 uia1total00;
array ui_spells (i) ui_spells80-ui_spells99 ui_spells00;   array ui2_spells (i) ui2_spells80-ui2_spells99
array ui1_spells (i) ui1_spells80-ui1_spells99 ui1_spells00; array wctotal (i) wctotal80-wctotal99 wctotal00;
array wc2total (i) wc2total80-wc2total99 wc2total00;       array wc1total (i) wc1total80-wc1total99 wc1total00;

```

```

array wcatotal (i) wcatotal80-wcatotal99 wcatotal00;      array wca2total (i) wca2total80-wca2total99 wca2total00;
array wcaltotal (i) wcaltotal80-wcaltotal99 wcaltotal00; array nu (i) nu80-nu99 nu00;

do i=1 to 21; nu=0; end;

do i=1 to 21;
/*if r2 interview, use r2 created var*/
/*if noninterview in r2, then use r1 created var*/
/*if variable not in r1 or r2 then set to 0*/
gatotal=ga2total;
  if ga2total=-5 then do; gatotal=gatotal; end;
  if gatotal=. then gatotal=0;
uitotal=ui2total;
  if ui2total=-5 then do; uitotal=uitotal; end;
  if uitotal=. then uitotal=0;
uiatotal=uia2total;
  if uia2total=-5 then do; uiatotal=uiatotal; end;
  if uiatotal=. then uiatotal=0;
ui_spells=ui2_spells;
  if ui2_spells=-5 then do; ui_spells=ui1_spells; end;
  if ui_spells=. then ui_spells=0;
wctotal=wc2total;
  if wc2total=-5 then do; wctotal=wctotal; end;
  if wctotal=. then wctotal=0;
wcatotal=wca2total;
  if wca2total=-5 then do; wcatotal=wcatotal; end;
  if wcatotal=. then wcatotal=0;
end;

wc_ever=wc2_ever;
  if wc2_ever=-5 then do; wc_ever=wc1_ever; end;
  if wc_ever=-4 then do; wc_ever=0; end;
ui_ever=ui2_ever;
  if ui2_ever=-5 then do; ui_ever=ui1_ever; end;
  if ui_ever=-4 then do; ui_ever=0; end;
gprg_ever=g2prg_ever;
  if g2prg_ever=-5 then do; gprg_ever=g1prg_ever; end;
  if gprg_ever=-4 then do; gprg_ever=0; end;

ack=0; ackm=0; ackh=0; fck=0; fckm=0; fckh=0; wck=0; wckm=0;
wckh=0; ock=0; ockm=0; ockh=0; uick=0; uickm=0; wcck=0; wcckm=0;

cm14=((bdate_y+14)-1980)*12+bdate_m;
cmb=((bdate_y)-1980)*12+bdate_m;
ym14=(BDATE_Y*100+BDATE_M)+1400;
iyM=YINT_Y*100+YINT_M;

/*not interviewed in round 2*/
if (p1210=1 and int98_y=-5) then do;
  dliyM=int97_y*100+(int97_m+1); dlicM=(int97_y-1980)*12+(int97_m+1); li_day=int97_d; end;
/*interviewed in round 2*/
if (p1210=1 and int98_y>-5) then do;
  dliyM=int98_y*100+(int98_m+1); dlicM=(int98_y-1980)*12+(int98_m+1); li_day=int98_d; end;
if p1210 ne 1 then dliyM=ym14;
if p1210 ne 1 then dlicM=cm14;

```

```

doicm=(YINT_Y-1980)*12+YINT_M;
aaicm=(YINT_Y-BDATE_Y)*12+YINT_M-BDATE_M;

//****************************************************************************
***/
array ui (l) ui001-ui245;           array uiamt (l) uiamt001-uiamt245;
array wc (l) wc001-wc245;           array wcamt (l) wcamt001-wcamt245;
array f (l) f001-f245;             array famt (l) famt001-famt245;
array fhlm (l) fhlm001-fhhm245;    array w (l) w001-w245;
array wamt (l) wamt001-wamt245;   array whhm (l) whhm001-whhm245;
array a (l) a001-a245;             array aamt (l) aamt001-aamt245;
array ahhm (l) ahhm001-ahhm245;   array o (l) o001-o245;
array oamt (l) oamt001-oamt245;  array ohhm (l) ohhm001-ohhm245;

/*2. [b1] begin - Rs eligible for a program in round 2*/
if P1210>-4 then do;
  if PS35707>-3 then P16200=PS35707;          /**AFDC**/
  if PS16377>-3 then P18900=PS16377;          /**Food Stamps**/
  if PS36087>-3 then P30500=PS36087;          /**Other**/
  if PS1897>-3 then P3500=PS1897;              /**Unemployment Compensation**/
  if PS21397>-3 then P21500=PS21397;          /**WIC**/
  if PS9867>-3 then P9700=PS9867;              /**Worker's Compensation**/

  do l=1 to 245;
    if cmb le L le doicm then do;
      a=0; w=0; f=0; o=0; wc=0; ui=0;
      end;
    end;

/** start YEAR information **/
array ysu1      (j) PS40001Y  PS40002Y  PS40003Y;  /**unemployment, sdli**/
array ysu2      (j) P40001Y  P40002Y  P40003Y;  /**unemployment, int**/
array ysc1      (j) PS102001Y PS102002Y PS102003Y; /**workers comp, sdil**/
array ysc2      (j) P102001Y  P102002Y P102003Y; /**workers comp, int**/
array ysf1      (j) PS191001Y PS191002Y PS191003Y; /**food stamps, sdli**/
array ysf2      (j) P194001Y  P194002Y P194003Y; /**food stamps, int**/
array ysw1      (j) PS220001Y PS220002Y PS220003Y; /**wic, sdli**/
array ysw2      (j) P220001Y  P220002Y P220003Y; /**wic, int**/
array ysa1      (j) PS358001Y PS358002Y PS358003Y; /**afdc, sdli**/
array ysa2      (j) P167001Y  P167002Y P167003Y; /**afdc, int**/
array yso1      (j) PS310001Y PS310002Y PS310003Y; /**other, sdli**/
array yso2      (j) P310001Y  P310002Y P310003Y; /**other, int**/

/** start MONTH information **/
array msu1      (j) PS40001M  PS40002M  PS40003M;  /**unemployment, sdli**/
array msu2      (j) P40001M  P40002M  P40003M;  /**unemployment, int**/
array msc1      (j) PS102001M PS102002M PS102003M; /**workers comp, sdil**/
array msc2      (j) P102001M  P102002M P102003M; /**workers comp, int**/
array msf1      (j) PS191001M PS191002M PS191003M; /**food stamps, sdli**/
array msf2      (j) P194001M  P194002M P194003M; /**food stamps, int**/
array msw1      (j) PS220001M PS220002M PS220003M; /**wic, sdli**/
array msw2      (j) P220001M  P220002M P220003M; /**wic, int**/
array msa1      (j) PS358001M PS358002M PS358003M; /**afdc, sdli**/
array msa2      (j) P167001M  P167002M P167003M; /**afdc, int**/
array mso1      (j) PS310001M PS310002M PS310003M; /**other, sdli**/
array mso2      (j) P310001M  P310002M P310003M; /**other, int**/

```

```

/**start DAY information for wc or ui only */
array dsu1      (j) PS42001    PS42002    PS42003;    /**unemployment, sdli*/
array esu1      (j) PS43001    PS43002    PS43003;    /**estimated unemployment, sdli*/
array dsu2      (j) P42001     P42002     P42003;    /**unemployment, int*/
array esu2      (j) P43001     P43002     P43003;    /**estimated unemployment, sdli*/
array dsc1      (j) PS104001   PS104002   PS104003;  /**workers comp, sdil*/
array esc1      (j) PS105001   PS105002   PS105003;  /**estimated workers comp, sdil*/
array dsc2      (j) P104001    P104002    P104003;  /**workers comp, int*/
array esc2      (j) P105001    P105002    P105003;  /**estimated workers comp, sdil*/

/** end YEAR information */
array yeu1      (j) PS47001Y   PS47002Y   PS47003Y;  /**unemployment, sdli*/
array yeu2      (j) P47001Y   P47002Y   P47003Y;  /**unemployment, int*/
array yec1      (j) PS109001Y  PS109002Y  PS109003Y; /**workers comp, sdil*/
array yec2      (j) P109001Y  P109002Y  P109003Y; /**workers comp, int*/
array yef1      (j) PS200001Y  PS200002Y  PS200003Y; /**food stamps, sdli*/
array yef2      (j) P200001Y  P200002Y  P200003Y; /**food stamps, int*/
array yew1      (j) PS226001Y  PS226002Y  PS226003Y; /**wic, sdli*/
array yew2      (j) P226001Y  P226002Y  P226003Y; /**wic, int*/
array yea1      (j) PS358401Y  PS358402Y  PS358403Y; /**afdc, sdli*/
array yea2      (j) P172001Y  P172002Y  P172003Y; /**afdc, int*/
array yeo1      (j) PS316001Y  PS316002Y  PS316003Y; /**other, sdli*/
array yeo2      (j) P316001Y  P316002Y  P316003Y; /**other, int*/

/** end MONTH information */
array meu1      (j) PS47001M   PS47002M   PS47003M;  /**unemployment, sdli*/
array meu2      (j) P47001M   P47002M   P47003M;  /**unemployment, int*/
array mec1      (j) PS109001M  PS109002M  PS109003M; /**workers comp, sdil*/
array mect2     (j) P109001M  P109002M  P109003M; /**workers comp, int*/
array mef1      (j) PS200001M  PS200002M  PS200003M; /**food stamps, sdli*/
array mef2      (j) P200001M  P200002M  P200003M; /**food stamps, int*/
array mew1      (j) PS226001M  PS226002M  PS226003M; /**wic, sdli*/
array mew2      (j) P226001M  P226002M  P226003M; /**wic, int*/
array meal1     (j) PS358401M  PS358402M  PS358403M; /**afdc, sdli*/
array meal2     (j) P172001M  P172002M  P172003M; /**afdc, int*/
array meo1      (j) PS316001M  PS316002M  PS316003M; /**other, sdli*/
array meo2      (j) P316001M  P316002M  P316003M; /**other, int*/

/**end DAY information for wc or ui only */
array deu1      (j) PS49001    PS49002    PS49003;    /**unemployment, sdli*/
array eeu1      (j) PS50001    PS50002    PS50003;    /**estimated unemployment, sdli*/
array deu2      (j) P49001     P49002     P49003;    /**unemployment, int*/
array eeu2      (j) P50001     P50002     P50003;    /**estimated unemployment, sdli*/
array dec1      (j) PS111001   PS111002   PS111003;  /**workers comp, sdil*/
array eec1      (j) PS112001   PS112002   PS112003;  /**estimated workers comp, sdil*/
array dec2      (j) P111001    P111002    P111003;  /**workers comp, int*/
array eec2      (j) P112001    P112002    P112003;  /**estimated workers comp, sdil*/

array ymsu1     (j) ymsu11-ymsu13;
array ymsc1     (j) ymsc11-ymsc13;
array ymsf1     (j) yms10 yms11 ymsf13;
array ymsw1     (j) yms05-yms07;
array ymsa1     (j) yms01-yms03;
array ymso1     (j) yms15 yms16 ymso13;

array ymsu2     (j) ymsu21-ymsu23;
array ymsc2     (j) ymsc21-ymsc23;
array ymsf2     (j) yms12-yms14;
array ymsw2     (j) yms08 yms09 ymsw23;
array ymsa2     (j) yms04 ymsa22 ymsa23;
array ymso2     (j) yms17 yms18 ymso23;

```

Appendix 5: Income and Assets Variable Creation

```

array csmu1  (j) csmu11-csmu13;          array csmu2  (j) csmu21-csmu23;
array csmc1  (j) csmc11-csmc13;          array csmc2  (j) csmc21-csmc23;
array csmf1  (j) csm10 csm11 csmf13;      array csmf2  (j) csm12-csm14;
array csmw1  (j) csm05-csm07;            array csmw2  (j) csm08 csm09 csmw23;
array csma1  (j) csm01-csm03;            array csma2  (j) csm04 csma22 csma23;
array csmo1  (j) csm15 csm16 csmo13;      array csmo2  (j) csm17 csm18 csmo23;

array ymeu1  (j) ymeu11-ymeu13;          array ymeu2  (j) ymeu21-ymeu23;
array ymec1  (j) ymec11-ymec13;          array ymec2  (j) ymec21-ymec23;
array ymef1  (j) yme10 yme11 ymef13;      array ymef2  (j) yme12-yme14;
array ymew1  (j) yme05-yme07;            array ymew2  (j) yme08 yme09 ymew23;
array ymeal  (j) yme01-yme03;            array ymea2  (j) yme04 ymea22 ymea23;
array ymeo1  (j) yme15 yme16 ymeo13;      array ymeo2  (j) yme17 yme18 ymeo23;

array cemu1  (j) cemu11-cemu13;          array cemu2  (j) cemu21-cemu23;
array cemc1  (j) cemc11-cemc13;          array cemc2  (j) cemc21-cemc23;
array cemf1  (j) cem10 cem11 cemf13;      array cemf2  (j) cem12-cem14;
array cemw1  (j) cem05-cem07;            array cemw2  (j) cem08 cem09 cemw23;
array cema1  (j) cem01-cem03;            array cema2  (j) cem04 cema22 cema23;
array cemo1  (j) cem15 cem16 cemo13;      array cemo2  (j) cem17 cem18 cemo23;

/** CURRENTLY receiving information */
array curu1  (j) PS44001    PS44002          PS44003;   /**unemployment, sdli**/
array curu2  (j) P44001     P44002          P44003;   /**unemployment, int**/
array curc1  (j) PS106001   PS106002         PS106003;  /**workers comp, sdil**/
array curc2  (j) P106001   P106002         P106003;  /**workers comp, int**/
array curf1  (j) PS21100A1  PS21100A2        PS21100A3; /**food stamps, no gap sdli**/
array curf2  (j) P196001   P196002         P196003;  /**food stamps, int**/
array curw1  (j) PS23700A1  PS23700A2        PS23700A3; /**wic, no gap sdli**/
array curw2  (j) P222001   P222002         P222003;  /**wic, int**/
array cura1  (j) PS359501   PS359502         PS359503;  /**afdc, no gap sdli**/
array cura2  (j) P169001   P169002         P169003;  /**afdc, int**/
array curo1  (j) PS32700A1  PS32700A2        PS32700A3; /**other, no gap sdli**/
array curo2  (j) P312001   P312002         P312003;  /**other, int**/

/** EDIT FLAGS for dates */
array eflagu1 (j) eflagu11-eflagu13;      /**unemployment, sdli**/
array eflagu2 (j) eflagu21-eflagu23;      /**unemployment, int**/
array eflagc1 (j) eflagc11-eflagc13;      /**workers comp, sdil**/
array eflagc2 (j) eflagc21-eflagc23;      /**workers comp, int**/
array eflagf1 (j) eflag10 eflag11 eflagf13;  /**food stamps, sdli**/
array eflagf2 (j) eflag12-eflag14;        /**food stamps, int**/
array eflagw1 (j) eflag05-eflag07;        /**wic, sdli**/
array eflagw2 (j) eflag08 eflag09 eflagw23;  /**wic, int**/
array eflaga1 (j) eflag01-eflag03;        /**afdc, sdli**/
array eflaga2 (j) eflag04 eflaga22 eflaga23;  /**afdc, int**/
array eflago1 (j) eflag15 eflag16 eflago13;  /**other, sdli**/
array eflago2 (j) eflag17 eflag18 eflago23;  /**other, int**/

/** EDIT FLAGS for amounts */
array aflagu1 (j) aflagu11-aflagu13;      /**unemployment, sdli**/
array aflagu2 (j) aflagu21-aflagu23;      /**unemployment, int**/
array aflagc1 (j) aflagc11-aflagc13;      /**workers comp, sdil**/
array aflagc2 (j) aflagc21-aflagc23;      /**workers comp, int**/
array aflagf1 (j) aflag10 aflag11 aflagf13;  /**food stamps, sdli**/
array aflagf2 (j) aflag12-aflag14;        /**food stamps, int**/

```

```

array aflagw1 (j) aflag05-aflag07;      /**wic, sdli**/
array aflagw2 (j) aflag08 aflag09 aflagw23; /**wic, int**/
array aflaga1 (j) aflag01-aflag03;      /**afdc, sdli**/
array aflaga2 (j) aflag04 aflaga22 aflaga23; /**afdc, int**/
array aflago1 (j) aflag15 aflag16 aflago13; /**other, sdli**/
array aflago2 (j) aflag17 aflag18 aflago23; /**other, int**/

/** AMOUNT RECEIVED - actual & estimated **/
array wincu1 (j) PS59001    PS59002    PS59003;    /**unemployment, amount sdli**/
array wincu2 (j) P59001     P59002     P59003;    /**unemployment, amount int**/
array ewincu1 (j) PS60001    PS60002    PS60003;    /**unemployment, est amount sdli**/
array ewincu2 (j) P60001     P60002     P60003;    /**unemployment, est amount int**/
array wincc1 (j) PS121001   PS121002   PS121003;   /**workers comp, amount sdli**/
array wincc2 (j) P121001    P121002    P121003;   /**workers comp, amount int**/
array ewincc1 (j) PS122001   PS122002   PS122003;   /**workers comp, est amount sdli**/
array ewincc2 (j) P122001    P122002    P122003;   /**workers comp, est amount int**/
array incf1 (j) PS207001   PS207002   PS207003;   /**food stamps, amount sdli**/
array incf2 (j) P207001    P207002    P207003;   /**food stamps, amount int**/
array eincf1 (j) PS208001   PS208002   PS208003;   /**food stamps, est amount sdli**/
array eincf2 (j) P208001    P208002    P208003;   /**food stamps, est amount int**/
array incw1 (j) PS233001   PS233002   PS233003;   /**wic, amount sdli**/
array incw2 (j) P233001    P233002    P233003;   /**wic, amount int**/
array eincw1 (j) PS234001   PS234002   PS234003;   /**wic, est amount sdli**/
array eincw2 (j) P234001    P234002    P234003;   /**wic, est amount int**/
array inca1 (j) PS359001   PS359002   PS359003;   /**afdc, amount sdli**/
array inca2 (j) P181001    P181002    P181003;   /**afdc, amount int**/
array einca1 (j) PS359101   PS359102   PS359103;   /**afdc, est amount sdli**/
array einca2 (j) P182001    P182002    P182003;   /**afdc, est amount int**/
array inco1 (j) PS323001   PS323002   PS323003;   /**other, amount sdli**/
array inco2 (j) P323001    P323002    P323003;   /**other, amount int**/
array einc01 (j) PS324001   PS324002   PS324003;   /**other, est amount sdli**/
array einco2 (j) P324001    P324002    P324003;   /**other, est amount int**/

/** ESTIMATED WEEKS - unable to answer start or stop quex **/
array wksu1 (j) PS52001   PS52002   PS52003;   /**unemployment, est weeks sdli**/
array wksu2 (j) P52001    P52002    P52003;   /**unemployment, est weeks int**/
array wksc1 (j) PS114001   PS114002   PS114003;   /**workers comp, est weeks sdli**/
array wksc2 (j) P114001    P114002    P114003;   /**workers comp, est weeks int**/
array wksf1 (j) PS202001   PS202002   PS202003;   /**food stamps, est weeks sdli**/
array wksf2 (j) P202001    P202002    P202003;   /**food stamps, est weeks int**/
array wksw1 (j) PS228001   PS228002   PS228003;   /**wic, est weeks sdli**/
array wksw2 (j) P228001    P228002    P228003;   /**wic, est weeks int**/
array wksa1 (j) PS358601   PS358602   PS358603;   /**afdc, est weeks sdli**/
array wksa2 (j) P175001    P175002    P175003;   /**afdc, est weeks int**/
array wkso1 (j) PS318001   PS318002   PS318003;   /**other, est weeks sdli**/
array wkso2 (j) P318001    P318002    P318003;   /**other, est weeks int**/

arraymosu1 (j) mosu11-mosu13;      array mosu2 (j) mosu21-mosu23;
array mosc1 (j) mosc11-mosc13;      array mosc2 (j) mosc21-mosc23;
array mosf1 (j) mosf11-mosf13;      array mosf2 (j) mosf21-mosf23;
array mosw1 (j) mosw11-mosw13;      array mosw2 (j) mosw21-mosw23;
array mosa1 (j) mosa11-mosa13;      array mosa2 (j) mosa21-mosa23;
array moso1 (j) moso11-moso13;      array moso2 (j) moso21-moso23;

/** PERSON receiving amount, respondent **/
array rvrfl (j) PS2090011  PS2090021  PS2090031;  /**food stamps, sdli**/

```

Appendix 5: Income and Assets Variable Creation

```

array rcvrf2      (j) P2090011   P2090021   P2090031;  /**food stamps, int*/
array rcvrw1      (j) PS2350011  PS2350021  PS2350031;  /**wic, sdli*/
array rcvrw2      (j) P2350011   P2350021   P2350031;  /**wic, int*/
array rcvra1      (j) PS3592011  PS3592021  PS3592031;  /**afdc, sdli*/
array rcvra2      (j) P1830011   P1830021   P1830031;  /**afdc, int*/
array rcvro1      (j) PS3250011  PS3250021  S3250031;  /**other, sdli*/
array rcvro2      (j) P3250011   P3250021   P3250031;  /**other, int*/

/** PERSON receiving amount, spouse */
array rcvsf1      (j) PS2090012   PS2090022  PS2090032;  /**food stamps, sdli*/
array rcvsf2      (j) P2090012   P2090022   P2090032;  /**food stamps, int*/
array rcvsw1      (j) PS2350012   PS2350022  PS2350032;  /**wic, sdli*/
array rcvsw2      (j) P2350012   P2350022   P2350032;  /**wic, int*/
array rcvsal      (j) PS3592012   PS3592022  PS3592032;  /**afdc, sdli*/
array rcvsa2      (j) P1830012   P1830022   P1830032;  /**afdc, int*/
array rcvso1      (j) PS3250012   PS3250022  PS3250032;  /**other, sdli*/
array rcvso2      (j) P3250012   P3250022   P3250032;  /**other, int*/

/** PERSON receiving amount, child */
array rcvcf1      (j) PS2090013   PS2090023  PS2090033;  /**food stamps, sdli*/
array rcvcf2      (j) P2090013   P2090023   P2090033;  /**food stamps, int*/
array rcvcw1      (j) PS2350013   PS2350023  PS2350033;  /**wic, sdli*/
array rcvcw2      (j) P2350013   P2350023   P2350033;  /**wic, int*/
array rcvca1      (j) PS3592013   PS3592023  PS3592033;  /**afdc, sdli*/
array rcvca2      (j) P1830013   P1830023   P1830033;  /**afdc, int*/
array rcvco1      (j) PS3250013   PS3250023  PS3250033;  /**other, sdli*/
array rcvco2      (j) P3250013   P3250023   P3250033;  /**other, int*/

/** PERSON receiving amount family member */
array rcvff1      (j) PS2090014   PS2090024  PS2090034;  /**food stamps, sdli*/
array rcvff2      (j) P2090014   P2090024   P2090034;  /**food stamps, int*/
array rcvfw1      (j) PS2350014   PS2350024  PS2350034;  /**wic, sdli*/
array rcvfw2      (j) P2350014   P2350024   P2350034;  /**wic, int*/
array rcvfa1      (j) PS3592014   PS3592024  PS3592034;  /**afdc, sdli*/
array rcvfa2      (j) P1830014   P1830024   P1830034;  /**afdc, int*/
array rcvfo1      (j) PS3250014   PS3250024  PS3250034;  /**other, sdli*/
array rcvfo2      (j) P3250014   P3250024   P3250034;  /**other, int*/

/** PERSON receiving amount, other person */
array rcvof1      (j) PS2090015   PS2090025  PS2090035;  /**food stamps, sdli*/
array rcvof2      (j) P2090015   P2090025   P2090035;  /**food stamps, int*/
array rcvow1      (j) PS2350015   PS2350025  PS2350035;  /**wic, sdli*/
array rcvow2      (j) P2350015   P2350025   P2350035;  /**wic, int*/
array rcvoa1      (j) PS3592015   PS3592025  PS3592035;  /**afdc, sdli*/
array rcvoa2      (j) P1830015   P1830025   P1830035;  /**afdc, int*/
array rcvo01      (j) PS3250015   PS3250025  PS3250035;  /**other, sdli*/
array rcvo02      (j) P3250015   P3250025   P3250035;  /**other, int*/

array whof1       (j) who10 who11 whof13;    array whof2      (j) who12-who14;
array whow1       (j) who05-who07;          array whow2      (j) who08 who09 whow23;
array whoa1       (j) who01-who03;          array whoa2      (j) who04 whoa22 whoa23;
array whoo1       (j) who15 who16 whoo13;    array whoo2      (j) who17 who18 whoo23;

array dlif1       (j) PS19063A1   PS19063A2   PS19063A3;
array dliw1       (j) PS21900A1   PS21900A2   PS21900A3;
array dia1        (j) PS357901   PS357902   PS357903;

```

```

array dlio1      (j) PS30900A1   PS30900A2   PS30900A3;
array dliui1     (j) PS18301    PS18302    PS19303;
array dliwc1     (j) PS98001    PS98002    PS98003;

/** these arrays encompass all programs for the arrays */
array ys          (k) ysa1 ysa2 ysw1 ysw2 ysf1 ysf2 yso1 yso2 ysu1 ysu2 ysc1 ysc2;
array ms          (k) msa1 msa2 msw1 msw2 msf1 msf2 mso1 mso2 msu1 msu2 msc1 msc2;
array ye          (k) yea1 yea2 yew1 yew2 yef1 yef2 yeo1 yeo2 yeu1 yeu2 yec1 yec2;
array me          (k) mea1 mea2 mew1 mew2 maf1 maf2 meo1 meo2 meu1 meu2 mec1 mec2;
array cur         (k) cura1 cura2 curw1 curw2 curf1 curf2 curo1 curo2 curu1 curu2 curc1 curc2;
array eflag       (k) eflaga1 eflaga2 eflagw1 eflagw2 eflagf1 eflagf2 eflago1 eflago2 eflagu1 eflagu2 eflagc1
eflagc2;
array aflag       (k) aflaga1 aflaga2 aflagw1 aflagw2 aflagf1 aflagf2 aflago1 aflago2 aflagu1 aflagu2 aflagc1
aflagc2;
array yms         (k) ymsa1 ymsa2 ymsw1 ymsw2 ymsf1 ymsf2 ymso1 ymso2 ymsu1 ymsu2 ymsc1 ymsc2;
array wks         (k) wksa1 wksa2 wksw1 wksw2 wksf1 wksf2 wkso1 wkso2 wksu1 wksu2 wksc1 wksc2;
array cwks        (k) PS35740 P16200 PS21430 P21500 PS16410 P18900 PS36120 P30500 PS1930 P3500 PS9900
P9700;
array yme         (k) ymea1 ymea2 ymew1 ymew2 ymef1 ymef2 ymeo1 ymeo2 ymeu1 ymeu2 ymec1 ymec2;
array mos         (k) mos1 mos2 mosw1 mosw2 mosf1 mosf2 moso1 moso2 mosu1 mosc1 mosc2;
array csm         (k) csma1 csma2 csmw1 csmw2 csmf1 csmf2 csmo1 csmo2 csmu1 csmu2 csmc1 csmc2;
array cem         (k) cema1 cema2 cemw1 cemw2 cemf1 cemf2 cemo1 cemo2 cemu1 cemu2 cemc1 cemc2;
array rcvr        (k) rcvra1 rcvra2 rcvrw1 rcvrw2 rcvrf1 rcvrf2 rcvro1 rcvro2 rcvr09-rcvr12;
array rcvs        (k) rcvs1 rcvs2 rcvsw1 rcvsw2 rcvsf1 rcvsf2 rcvso1 rcvso2 rcvs09-rcvs12;
array rcvc        (k) revca1 revca2 revcw1 revcw2 revcf1 revcf2 revco1 revco2 revc09-rcvc12;
array rcvf        (k) revfa1 revfa2 rcvfw1 rcvfw2 rcvff1 rcvff2 rcvfo1 rcvfo2 rcvf09-rcvf12;
array revo        (k) rcoa1 rcoa2 rcvow1 rcvow2 rcvof1 rcvof2 rcvo1 rcvo2 rcvo09-rcvo12;
array who         (k) whoa1 whoa2 whow1 whow2 whof1 whof2 whoo1 whoo2 who09-who12;
array dli         (k) dlia1 dlia2 dliw1 dliw2 dlif1 dlif2 dlio1 dlio2 dliui1 dliui2 dliwc1 dliwc2;
array inc         (k) inca1 inca2 incw1 incw2 incf1 incf2 inco1 inco2 wincu1 wincu2 wincc1 wincc2;

array agefla1     (j) agefla11 agefla12 agefla13;           array agefla2     (j) agefla21 agefla22 agefla23;
array ageflw1     (j) ageflw11 ageflw12 ageflw13;           array ageflw2     (j) ageflw21 ageflw22 ageflw23;
array ageflf1     (j) ageflf11 ageflf12 ageflf13;           array ageflf2     (j) ageflf21 ageflf22 ageflf23;
array ageflo1     (j) ageflo11 ageflo12 ageflo13;           array ageflo2     (j) ageflo21 ageflo22 ageflo23;
array ageflu1     (j) ageflu11 ageflu12 ageflu13;           array ageflu2     (j) ageflu21 ageflu22 ageflu23;
array ageflc1     (j) ageflc11 ageflc12 ageflc13;           array ageflc2     (j) ageflc21 ageflc22 ageflc23;
array agefl      (k) agefla1 agefla2 ageflw1 ageflw2 ageflf1 ageflf2 ageflo1 ageflo2 ageflu1 ageflc1 ageflc2;

/*arrays to determine days in month that need to be accounted for in wc and ui arrays*/
array st_year     (z) ysu1 ysu2 ysc1 ysc2;           array st_mo       (z) msu1 msu2 msc1 msc2;
array st_day      (z) dsu1 dsu2 dsc1 dsc2;           array est_day     (z) esu1 esu2 esc1 esc2;
array en_year     (z) yeu1 yeu2 yec1 yec2;           array en_mo       (z) meu1 meu2 mec1 mec2;
array en_day      (z) deu1 deu2 dec1 dec2;           array een_day     (z) eeu1 eeu2 eec1 eec2;
array s_day       (z) su1_d su2_d sw1_d sw2_d;           array e_day       (z) eu1_d eu2_d ew1_d ew2_d;
array dli_day     (z) dliui1 dliui2 dliwc1 dliwc2;           array cur_day     (z) curu1 curu2 curc1 curc2;
array est_wks     (z) wksu1 wksu2 wksc1 wksc2;

/*****************************************/
*to determine the length of a spell of unemployment or workers compensation determine days;
do z=1 to 4;
do j=1 to 3;
  if st_day>0 or est_day>0 then do;           /*set start day, if given*/
    if st_mo ne en_mo or st_year ne en_year then do;
      if st_mo=1 or st_mo=3 or st_mo=5 or st_mo=7 or st_mo=8 or st_mo=10 or st_mo=12 then do;
        if st_day>0 and st_day ne . then s_day=(32-st_day);

```

```

if est_day=1 then s_day=31; if est_day=2 then s_day=21; if est_day=3 then s_day=11;
  if dli_day=1 then s_day=(32-li_day);
end;
if st_mo=4 or st_mo=6 or st_mo=9 or st_mo=11 then do;
  if st_day>0 and st_day ne . then s_day=(31-st_day);
  if est_day=1 then s_day=30; if est_day=2 then s_day=20; if est_day=3 then s_day=10;
  if dli_day=1 then s_day=(31-li_day);
end;
if st_mo=2 then do;
  if st_day>0 and st_day ne . then s_day=(29-st_day);
  if est_day=1 then s_day=28; if est_day=2 then s_day=18; if est_day=3 then s_day=8;
  if dli_day=1 then s_day=(29-li_day);
end;
end;
if st_year>-4 and st_year<0 then do; /*start date not given*/
  if dli_day=1 and est_wks>4 then s_day=(32-li_day);
  if dli_day=1 and est_wks>0 and est_wks<5 then s_day=(est_wks*7);
  if dli_day ne 1 and est_wks>4 then s_day=1;
  if dli_day ne 1 and est_wks>0 and est_wks<5 then s_day=(est_wks*7);
end;
if en_day>0 and en_day ne . then do; e_day=en_day; end; /*set end day, if given*/
if en_day<0 and een_day>0 and een_day ne . then do; /*end day unknown, estimate given*/
  if een_day=1 then e_day=9; if een_day=2 then e_day=19; if een_day=3 then e_day=28;
end;
if cur_day=1 then do; e_day=yint_d; end; /*continuous - end day is int date*/
if st_mo=en_mo and st_year=en_year and st_day le en_day and st_day>0 and en_day>0 then do;
  s_day=(en_day-st_day+1); e_day=0;
end;
if ((st_year>-4 and st_year<0) or (en_year>-4 and en_year<0)) and cur_day ne 1 then do; e_day=31; end;
end;
end;

/*********************  
*****
This portion of the SAS program defines the start date and end dates. If the respondent reports still receiving, the interview date is used as the temporary end date for the last loop reported. In the next survey round, the respondent will be asked if he or she is still receiving and, if not, a permanent end date equivalent to the interview date of the previous round will be assigned. Users will be able to tell which method was used by looking at the following participation flag variable created during the program. The categories are the following:
  1=respondent reported participation dates      5=stop month and year imputed
  2=start month imputed                         6=start and stop dates imputed
  3=start month and year imputed                7=error in data due to round 2 programming error
  4=stop month imputed
*****/
```

```

do k=1 to 12; /*[b2]*/
  do j=1 to 3; /*[b3]*/
    if ys>0 and ms>0 then yms=(ys*100)+ms; /*set start date if both start month and year are valid*/
    if dli=1 then yms=dliy; /*set start date to date last int if receiving then*/
    if cur=1 then yme=iym; /*set end date to current int date if receiving now*/
    if ye>0 and me>0 then yme=(ye*100)+me; /*set end date if both end month and year are valid*/
    if yme>iym then yme=iym; /*if end month greater than int month, constrain by int month*/
    if yms>0 and yme ge dliy then eflag=1;
    if wks ge 0 and (ys<0 or ms<0 or ye<0 or me<0) then do;
      mos=floor(wks/4.3); /*determine months from estimated weeks variable*/

```

```

end;

agefl=0;
if ys>0 and ys<1990 then agefl=1;

*****1. if start year is known and month is unknown *****
 $\text{1.a'}$  If weeks are known and currently receiving, then count backwards by the number of weeks from the interview date. If the number of weeks falls short of the start year, the start month is December of that year. If the number of weeks is past the start year, then the start month is January of that year.****/
if ys>0 and -3 le ms le -1 then do;
  if (wks>0 and cwks=1) then do;
    yme=iym;
    if mos le YINT_M then yms=iym-mos;
    if YINT_M le mos le (YINT_M+12) then yms=iym-100+(mos-12);
    if (YINT_M+12) le mos le (YINT_M+24) then yms=iym-200+(mos-24);
    if (YINT_M+24) le mos le (YINT_M+36) then yms=iym-300+(mos-36);
    if (YINT_M+36) le mos le (YINT_M+48) then yms=iym-400+(mos-48);
    if (YINT_M+48) le mos le (YINT_M+60) then yms=iym-500+(mos-60);
    if (YINT_M+60) le mos le (YINT_M+72) then yms=iym-600+(mos-72);
    if (YINT_M+72) le mos le (YINT_M+84) then yms=iym-700+(mos-84);
    if (YINT_M+84) le mos le (YINT_M+96) then yms=iym-800+(mos-96);
    if (YINT_M+96) le mos le (YINT_M+108) then yms=iym-900+(mos-108);
    if (YINT_M+108) le mos le (YINT_M+120) then yms=iym-1000+(mos-120);
    if yms<((ys*100)+01) then yms=((ys*100)+01);
    if yms<dliym then yms=dliym;
    if yms>((ys*100)+12) then yms=((ys*100)+12);
  end;

/*weeks missing & continuously receiving-set end date to current interview date and start date to December of start year*/
else if (-3 le wks le 0 and cwks=1) then do; yme=iym; yms=((ys*100)+12); if yms>iym then yms=iym; end;

/*weeks missing and not continuously receiving - set end date to December and start date to January of start year*/
else if (-3 le wks le 0 and cwks=0) then do;
  yms=((ys*100)+01); yme=((ys*100)+12); if yme>iym then yme=iym;
end;

 $\text{1.a''}$  if weeks are known and not currently receiving, then count forward by the number of weeks from January of the start year. If the count exceeds the interview date then stop counting at the interview date.****/
else if (wks>0 and cwks=0) then do;
  yms=((ys*100)+01);
  if yms<dliym then yms=dliym;
  if 0 le mos le 12 then yme=(yms+mos);
  if wks>0 and cwks=0 and 13 le mos le 24 then yme=((ys+1)*100)+mos-12;
  if wks>0 and cwks=0 and 25 le mos le 36 then yme=((ys+2)*100)+mos-24;
  if wks>0 and cwks=0 and 37 le mos le 48 then yme=((ys+3)*100)+mos-36;
  if wks>0 and cwks=0 and 49 le mos le 60 then yme=((ys+4)*100)+mos-48;
  if wks>0 and cwks=0 and 61 le mos le 72 then yme=((ys+5)*100)+mos-60;
  if wks>0 and cwks=0 and 73 le mos le 84 then yme=((ys+6)*100)+mos-72;
  if wks>0 and cwks=0 and 85 le mos le 96 then yme=((ys+7)*100)+mos-84;
  if wks>0 and cwks=0 and 97 le mos le 108 then yme=((ys+8)*100)+mos-96;
  if wks>0 and cwks=0 and 109 le mos le 120 then yme=((ys+9)*100)+mos-108;
  if yme>iym then yme=iym;
end;
if yme>0 and cwks=0 then yms=((ys*100)+01);
if yms<dliym then yms=dliym; eflag=2;

```

end;

*******2. if start year is unknown but weeks are known then count back from interview date if currently receiving.** If not currently receiving, then count back from interview date to find the most recent year respondent could have begun receiving and receive for that number of months; then count forward the number of months from Jan of that year *****/

```

if -3 le ys le -1 then do;
  if (wks>0 and cwks=1) then do;
    if 0 le mos le YINT_M then yms=iym-mos;
    if YINT_M le mos le (YINT_M+12) then yms=iym-100-(mos-12);
    if (YINT_M+12) le mos le (YINT_M+24) then yms=iym-200-(mos-24);
    if (YINT_M+24) le mos le (YINT_M+36) then yms=iym-300-(mos-36);
    if (YINT_M+36) le mos le (YINT_M+48) then yms=iym-400-(mos-48);
    if (YINT_M+48) le mos le (YINT_M+60) then yms=iym-500-(mos-60);
    if (YINT_M+60) le mos le (YINT_M+72) then yms=iym-600-(mos-72);
    if (YINT_M+72) le mos le (YINT_M+84) then yms=iym-700-(mos-84);
    if (YINT_M+84) le mos le (YINT_M+96) then yms=iym-800-(mos-96);
    if (YINT_M+96) le mos le (YINT_M+108) then yms=iym-900-(mos-108);
    if (YINT_M+108) le mos le (YINT_M+120) then yms=iym-1000-(mos-120);
    yme=iym;
  end;
  if (-3 le wks le 0 and cwks=1) then do; yms=dliym; yme=iym; end;
  if (wks>0 and cwks=0) then do;
    if 0 le mos le YINT_M then do; yms=(YINT_Y*100)+01; yme=(YINT_Y*100)+01+mos; end;
    if (YINT_M+01) le mos le (YINT_M+12) then do;
      yms=((YINT_Y-1)*100)+01;
      if (YINT_M+01) le mos le 12 then yme=((YINT_Y-1)*100)+01+mos;
      if 13 le mos le (YINT_M+12) then yme=(YINT_Y*100)+01+mos;
    end;
    if (YINT_M+13) le mos le (YINT_M+24) then do;
      yms=((YINT_Y-2)*100)+01;
      if (YINT_M+13) le mos le 24 then yme=((YINT_Y-1)*100)+01+mos;
      if 25 le mos le (YINT_M+24) then yme=(YINT_Y*100)+01+mos;
    end;
    if yme>iym then yme=iym;
  end;
  eflag=3;
end;
if yme>0 and yms=. then do;
  yms=dliym; if j=1 then eflag=3; else if j>1 then eflag=7;
end;

```

*******3. If stop year is known and weeks are known and not currently receiving, but stop month is not known,** then count forward from start year. If the number of months falls short of the stop year, then use January of the end year as the stop date; if the number of months exceeds the stop year, then end the array in the December of the stop year. If the stop year is equal to the interview year and the stop month exceeds the interview month, then stop at the interview date. If currently receiving, use interview date as the stop date. *****/

```

if yms>0 and ye>0 and -2 le me le -1 then do;
  if wks>0 then do;
    if 01 le ((yms- round (yms,100))+mos) le 12 then yme=yms+mos;
    if 12 lt ((yms- round (yms,100))+mos) le 24 then yme=yms+100+mos-12;
    if 24 lt ((yms- round (yms,100))+mos) le 36 then yme=yms+200+mos-24;
    if yme>((ye*100)+12) then yme=((ye*100)+12);
    if yme<((ye*100)+01) then yme=((ye*100)+01);
    if yme>iym then yme=iym;
  end;

```

```

if -3 le wks le 0 then yme=((ye*100)+12);
if yme>iym then yme=iym;
eflag=4;
end;

**** 4. if stop year is unknown, and weeks are known****
if -3 le ye le -1 then do;
if yms>0 and wks>0 then do;
if 01 le ((yms- round (yms,100))+mos) le 12 then yme=yms+mos;
if 12 lt ((yms- round (yms,100))+mos) le 24 then yme=yms+100+mos-12;
if 24 lt ((yms- round (yms,100))+mos) le 36 then yme=yms+200+mos-24;
if yme>iym then yme=iym;
end;

**** 5. if stop year is unknown, and weeks are unknown****
if yms>0 and -3 le wks le 0 then yme=(round (yms,100))+12; eflag=5; end;

**** 6. if the start and stop years are unknown and the weeks are unknown, use current and last interview date****
if -3 le ys le -1 and cwks=0 and -3 le wks le 0 then do;
yme=iym; yms=dliym; if j=1 then eflag=6; else if j>1 then eflag=7;
end;

```

/*****

This portion of the program creates a variable that determines who receives afdc in the household. It collapses the answers to 8 categories. The coding is the following -

| | |
|---------------------------------|---|
| 1=respondent only | 5=respondent and child |
| 2=spouse/partner only | 6=spouse/partner and child |
| 3=child only | 7=respondent and spouse/partner and child |
| 4=respondent and spouse/partner | 8=other |

The first 7 categories may include an 'other' person as captured by response categories 4 and 5 in the original question. If only another person is listed as receiving, then the 8th answer category is used in the created variable.*****

```

if rcvr=1 and rcvs=0 and rcvc=0 then who=01;      if rcvr=0 and rcvs=1 and rcvc=0 then who=02;
if rcvr=0 and rcvs=0 and rcvc=1 then who=03;      if rcvr=1 and rcvs=1 and rcvc=0 then who=04;
if rcvr=1 and rcvs=0 and rcvc=1 then who=05;      if rcvr=0 and rcvs=1 and rcvc=1 then who=06;
if rcvr=1 and rcvs=1 and rcvc=1 then who=07;      if -3 le rcvr lt 0 then who=rcvr;
if rcvr=0 and rcvs=0 and (rcvf=1 or rcvo=1) then who=08;

```


This portion of the program uses the category reported by the respondent to create an estimated amount. The estimated amount is the midpoint rounded down. Note that the 12th category lists \$1251 as the amount. This amount was chosen since the category is unbounded - the number ***** represents one dollar above the lower bound.*****

```

if -3 le incf1 le -1 and 1 le eincf1 le 10 then incf1=(eincf1*100)-50;
if -3 le incf1 le -1 and eincf1=11 then incf1=1125;
if -3 le incf1 le -1 and eincf1=12 then incf1=1251;
if -3 le incf2 le -1 and 1 le eincf2 le 10 then incf2=(eincf2*100)-50;
if -3 le incf2 le -1 and eincf2=11 then incf2=1125;
if -3 le incf2 le -1 and eincf2=12 then incf2=1251;
if -3 le inca1 le -1 and 1 le einca1 le 10 then inca1=(einca1*100)-50;
if -3 le inca1 le -1 and einca1=11 then inca1=1125;

```

```

if -3 le inca1 le -1 and einca1=12 then inca1=1251;
if -3 le inca2 le -1 and 1 le einca2 le 10 then inca2=(einca2*100)-50;
if -3 le inca2 le -1 and einca2=11 then inca2=1125;
if -3 le inca2 le -1 and einca2=12 then inca2=1251;
if -3 le inco1 le -1 and 1 le einco1 le 10 then inco1=(einco1*100)-50;
if -3 le inco1 le -1 and einco1=11 then inco1=1125;
if -3 le inco1 le -1 and einco1=12 then inco1=1251;
if -3 le inco2 le -1 and 1 le einco2 le 10 then inco2=(einco2*100)-50;
if -3 le inco2 le -1 and einco2=11 then inco2=1125;
if -3 le inco2 le -1 and einco2=12 then inco2=1251;
if -3 le incw1 le -1 and 1 le eincw1 le 5 then incw1=(eincw1*20)-10;
if -3 le incw1 le -1 and eincw1=6 then incw1=101;
if -3 le incw2 le -1 and 1 le eincw2 le 5 then incw2=(eincw2*20)-10;
if -3 le incw2 le -1 and eincw2=6 then incw2=101;

if -3 le wincu1<0 le -1 and ewincu1=1 then wincu1=(50/7);
if -3 le wincu1<0 le -1 and ewincu1=1 then wincu1=(125/7);
if -3 le wincu1<0 le -1 and ewincu1=1 then wincu1=(175/7);
if -3 le wincu1<0 le -1 and ewincu1=1 then wincu1=(225/7);
if -3 le wincu1<0 le -1 and ewincu1=1 then wincu1=(275/7);
if -3 le wincu1<0 le -1 and ewincu1=1 then wincu1=(351/7);
if -3 le wincu2<0 le -1 and ewincu2=1 then wincu2=(50/7);
if -3 le wincu2<0 le -1 and ewincu2=1 then wincu2=(125/7);
if -3 le wincu2<0 le -1 and ewincu2=1 then wincu2=(175/7);
if -3 le wincu2<0 le -1 and ewincu2=1 then wincu2=(225/7);
if -3 le wincu2<0 le -1 and ewincu2=1 then wincu2=(275/7);
if -3 le wincu2<0 le -1 and ewincu2=1 then wincu2=(351/7);

if -3 le wincc1 le -1 and 1 le ewincc1 le 10 then wincc1=((ewincc1*100)-50)/7;
if -3 le wincc1 le -1 and ewincc1=11 then wincc1=(1125/7);
if -3 le wincc1 le -1 and ewincc1=12 then wincc1=(1251/7);
if -3 le wincc2 le -1 and 1 le ewincc2 le 10 then wincc2=((ewincc2*100)-50)/7;
if -3 le wincc2 le -1 and ewincc2=11 then wincc2=(1125/7);
if -3 le wincc2 le -1 and ewincc2=12 then wincc2=(1251/7);

if inc>-4 then aflag=0;

csm=(round(yms,100)-198000)*.12+(yms-round(yms,100));
cem=(round(yme,100)-198000)*.12+(yme-round(yme,100));
cdli=(round(dliym,100)-198000)*.12+(dliym-round(dliym,100))-1;      /**added to zero out people who
back report**/
```

******to compute the spells of ui;**

```

if 9 le K le 10 then do;
  if 1 le CSM le 12 then NU80=NU80+1;      if 13 le CSM le 24 then NU81=NU81+1;
  if 25 le CSM le 36 then NU82=NU82+1;      if 37 le CSM le 48 then NU83=NU83+1;
  if 49 le CSM le 60 then NU84=NU84+1;      if 61 le CSM le 72 then NU85=NU85+1;
  if 73 le CSM le 84 then NU86=NU86+1;      if 85 le CSM le 96 then NU87=NU87+1;
  if 97 le CSM le 108 then NU88=NU88+1;      if 109 le CSM le 120 then NU89=NU89+1;
  if 121 le CSM le 132 then NU90=NU90+1;      if 133 le CSM le 144 then NU91=NU91+1;
  if 145 le CSM le 156 then NU92=NU92+1;      if 157 le CSM le 168 then NU93=NU93+1;
  if 169 le CSM le 180 then NU94=NU94+1;      if 181 le CSM le 192 then NU95=NU95+1;
  if 193 le CSM le 204 then NU96=NU96+1;      if 205 le CSM le 216 then NU97=NU97+1;
  if 217 le CSM le 228 then NU98=NU98+1;      if 229 le CSM le 240 then NU99=NU99+1;
  if 241 le CSM le 245 then NU00=NU00+1;
end;
```

```

C=0;
do L=1 to 245; C=C+1;
if 0 le L le doicm then do;
  if csm LE C LE cem then do;
    if 1 le k le 2 then do; a=eflag; ahhm=who; aamt=inc; aodd=0;
      if aamt>1000 then do; aodd=aamt; end;
      if a>0 and (aamt=. or ahhm=.) then do; ack=ack+1; end;
      if aamt>0 and (a=0 or ahhm=.) then do; ackm=ackm+1; end;
      if ahhm>0 and (a=0 or aamt=.) then do; ackh=ackh+1; end;
      if aamt=0 then dummya=1;
    end;
    if 3 le k le 4 then do; w=eflag; whhm=who; wamt=inc; wodd=0;
      if wamt>1000 then do; wodd=wamt; end;
      if w>0 and (wamt=. or whhm=.) then do; wck=wck+1; end;
      if wamt>0 and (w=0 or whhm=.) then do; wckm=wckm+1; end;
      if whhm>0 and (w=0 or wamt=.) then do; wckh=wckh+1; end;
      if wamt=0 then dummyw=1;
    end;
    if 5 le k le 6 then do; f=eflag; fhdm=who; famt=inc; fodd=0;
      if famt>1000 then do; fodd=famt; end;
      if f>0 and (famt=. or fhdm=.) then do; fck=fck+1; end;
      if famt>0 and (f=0 or fhdm=.) then do; fckm=fckm+1; end;
      if fhdm>0 and (f=0 or famt=.) then do; fckh=fckh+1; end;
      if fhdm=. and f>0 and famt ne . then do; fhdm=-3; fckneg=1; end; /*this fixes a problem in round 3*/
      if famt=0 then dummyf=1;
    end;
    if 7 le k le 8 then do; o=eflag; ohhm=who; oamt=inc; oodd=0;
      if oamt>1000 then do; oodd=oamt; end;
      if o>0 and (oamt=. or ohhm=.) then do; ock=ock+1; end;
      if oamt>0 and (o=0 or ohhm=.) then do; ockm=ockm+1; end;
      if ohhm>0 and (o=0 or oamt=.) then do; ockh=ockh+1; end;
      if oamt=0 then dummyo=1;
    end;
  end;
end;
end;
end;

C=0;
do L=1 to 245; C=C+1;
if 0 le L le doicm then do;
  if csm LE C LE cem then do; do z=1 to 2;
    if 9 le k le 10 then do;
      if C=csm and inc>0 then do; ui=eflag; if s_day ne . then uiamt= round ((inc*s_day), 1); end;
      else if C=cem and inc>0 then do; ui=eflag; if e_day ne . then uiamt= round ((inc*s_day), 1); end;
      else if csm<C and C<cem and inc>0 then do; ui=eflag;
        if C in (169, 171, 173, 175, 176, 178, 180, 181, 183, 185, 187, 188, 190, 192, 193, 195, 197, 199, 200,
          202, 204, 205, 207, 209, 211, 212, 214, 216, 217, 219, 221, 223, 224, 226, 228, 229, 231, 233,
          235, 236, 238, 240, 241, 243) then do; uiamt= round ((inc*31),1); end;
        else if C in (172, 174, 177, 179, 184, 186, 189, 191, 196, 198, 201, 203, 208, 210, 213, 215, 220, 222,
          225, 227, 232, 234, 237, 239) then do; uiamt=round ((inc*30),1); end;
        else if C in (158,194,242) then do; uiamt=round ((inc*29),1); end;
        else if C in (170,182,206,218,230) then do; uiamt= round ((inc*28),1); end;
        else if inc<0 then do; ui=eflag; uiamt=inc; end;
      end;
      uiodd=0;
    end;
  end;

```

```

if uiamt>1000 then do; uiodd=uiamt; end;
if ui>0 and uiamt=. then do; uick=uick+1; end;
if uiamt>0 and ui=0 then do; uickm=uickm+1; end;
if uiamt=0 then dummyui=1;
end;
end;
end;
end;
end;

C=0;
do L=1 to 245; C=C+1;
if 0 le L le doicm then do;
  if csm LE C LE cem then do; do z=3 to 4;
    if 11 le k le 12 then do;
      if C=csm and inc>0 then do; wc=eflag; if s_day ne . then wcamt= round ((inc*s_day), 1); end;
      else if C=cem and inc>0 then do; wc=eflag; if e_day ne . then wcamt= round ((inc*e_day), 1); end;
      else if csm<C and C<cem and inc>0 then do; wc=eflag;
        if C in (169, 171, 173, 175, 176, 178, 180, 181, 183, 185, 187, 188, 190, 192, 193, 195, 197, 199, 200,
          202, 204, 205, 207, 209, 211, 212, 214, 216, 217, 219, 221, 223, 224, 226, 228, 229, 231, 233,
          235, 236, 238, 240, 241, 243) then do; wcamt= round ((inc*31),1); end;
        else if C in (172, 174, 177, 179, 184, 186, 189, 191, 196, 198, 201, 203, 208, 210, 213, 215, 220, 222,
          225, 227, 232, 234, 237, 239) then do; wcamt=round ((inc*30),1); end;
        else if C in (158,194,242) then do; wcamt=round ((inc*29),1); end;
        else if C in (170,182,206,218,230) then do; wcamt= round ((inc*28),1); end;
        else if inc<0 then do; wc=eflag; wcamt=inc; end;
      end;
      wcodd=0;
      if wcamt>1000 then do; wcodd=wcamt; end;
      if wc>0 and wcamt=. then do; wcck=wcck+1; end;
      if wcamt>0 and wc=0 then do; wcckm=wcckm+1; end;
      if wcamt=0 then dummywc=1;
    end;
  end;
end;
end;
end;
end;

/****(new r3) if start date<current int date and R went through the section previously, zero months post dli***/
if csm le cdli and P010 eq 1 and csm ne . then do;
  if csm le C le cdli then do;
    if 1 le k le 2 then do; a=0; ahhm=.; aamt=.; end;
    if 3 le k le 4 then do; w=0; whhm=.; wamt=.; end;
    if 5 le k le 6 then do; f=0; fhhm=.; famt=.; end;
    if 7 le k le 8 then do; o=0; ohhm=.; oamt=.; end;
    if 9 le k le 10 then do; ui=0; uiamt=.; end;
    if 11 le k le 12 then do; wc=0; wcamt=.; end;
  end;
end;
end;
end; /**[e3]*/
end; /**[e2]*/ */ /*2. [e1] end*/

```

/*3. begin - Rs eligible in round 1 or round 2: used to combine data */

| | |
|--|--|
| array ckm (i) ckm80-ckm99 ckm00; | array ckamt (i) ckamt80-ckamt99 ckamt00; |
| array M (i) M80-M99 M00; | array amt (i) amt80-amt99 amt00; |

```

array um      (i) um80-um99 um00;          array uamt     (i) uamt80-uamt99 uamt00;
array ckuiamt (i) ckuiamt80-ckuiamt99 ckuiamt00;  array wm       (i) wm80-wm99 wm00;
array camt    (i) camt80-camt99 camt00;        array ckwcamt (i) ckwcamt80-ckwcamt99 ckwcamt00;
array na      (i) na80-na99 na00;          array nn       (i) nn80-nn99 nn00;
array ua      (i) ua80-ua99 ua00;          array un       (i) un80-un99 un00;
array wa      (i) wa80-wa99 wa00;          array wn       (i) wn80-wn99 wn00;

do i=1 to 21;
  if gprg_ever>-4 or P1210>-4 or p1210=-5 then do;
    m=0; ttM=0; amt=0; UM=0; WM=0; TTLU=0; TTLW=0; camt=0; uamt=0; na=0; nn=0; ua=0; un=0; wa=0;
  wn=0;
    incprg=0; prgamt=0; nnp=0; nap=0; out=0;
  end;
end;

amnths=0;      /*initialize month counter for correction of problematic amounts reported*/
fmnths=0;      omnths=0;      wmnths=0;      uimnths=0;      wcmnths=0;

if gprg_ever>-4 or P1210>-4 or p1210=-5 then do;
  do L=1 to 245;
    /*calculate total no. months to divide the problematic amts over*/
    if a>0 and aamt>1000 and aodd>0 then do; amnths=amnths+1; end;
    if f>0 and famt>1000 and fodd>0 then do; fmnths=fmnths+1; end;
    if o>0 and oamt>1000 and oodd>0 then do; omnths=omnths+1; end;
    if w>0 and wamt>1000 and wodd>0 then do; wmnths=wmnths+1; end;
    if ui>0 and uiamt>1000 and uiodd>0 then do; uimnths=uimnths+1; end;
    if wc>0 and wcamt>1000 and wcodd>0 then do; wcmnths=wcmnths+1; end;
  end;
end;

if gprg_ever>-4 or P1210>-4 or p1210=-5 then do;
  do L=1 to 245;
    /*divide the amount by the number of months*/
    if aamt>1000 and amnths>0 and a>0 then do; aamt=round(aamt/amnths, 1); end;
    if famt>1000 and fmnths>0 and f>0 then do; famt=round(famt/fmnths, 1); end;
    if oamt>1000 and omnths>0 and o>0 then do; oamt=round(oamt/omnths, 1); end;
    if wamt>1000 and wmnths>0 and w>0 then do; wamt=round(wamt/wmnths, 1); end;
    if uiamt>1000 and uimnths>0 and ui>0 then do; uiamt=round(uiamt/uimnths, 1); end;
    if wcamt>1000 and wcmnths>0 and wc>0 then do; wcamt=round(wcamt/wcmnths, 1); end;

  if 1 le L le 12 then do;
    if M80 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M80=M80+1; end;
    if a>1 then M80=-3;      if w>1 then M80=-3;      if f>1 then M80=-3;      if o>1 then M80=-3;
    array allamt80 aamt wamt famt oamt;
    do over allamt80;
      if -3 le allamt80 le -1 then do; nn80=nn80+1; amt80=allamt80; end;
      if allamt80 ge 0 then do; na80=na80+1; if na80=0 and nn80=0 then amt80=allamt80;
        if na80>1 and nn80=0 then amt80=amt80+allamt80; if M80=-3 then amt80=-3; end;
    end;
    if UM80 ge 0 and UI>0 then do; UM80=UM80+1; end;
    if UI>1 then UM80=-3;
    if -3 le uiamt le -1 then do; un80=un80+1; uamt80=uiamt; end;
    if uiamt ge 0 then do; ua80=ua80+1; if ua80=1 and un80=0 then UAMT80=UIAMT;
      if ua80>1 and un80=0 then uamt80=uamt80+uiamt; if UM80=-3 then uamt80=-3; end;
    if WM80 ge 0 and WC>0 then do; WM80=WM80+1; end;
    if WC>1 then WM80=-3;
  end;
end;

```

```

if -3 le wcamt le -1 then do; wn80=wn80+1; camt80=wcamt; end;
if wcamt ge 0 then do; wa80=wa80+1; if wa80=1 and wn80=0 then cAMT80=WCAMT;
    if wa80>1 and wn80=0 then camt80=camt80+wcamt; if WM80=-3 then camt80=-3; end;
end;

if 13 le L le 24 then do;
    if M81 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M81=M81+1; end;
    if a>1 then M81=-3;      if w>1 then M81=-3;      if f>1 then M81=-3;      if o>1 then M81=-3;
    array allamt81 aamt wamt famt oamt;
    do over allamt81;
        if -3 le allamt81 le -1 then do; nn81=nn81+1; amt81=allamt81; end;
        if allamt81 ge 0 then do; na81=na81+1; if na81=0 and nn81=0 then amt81=allamt81;
            if na81>1 and nn81=0 then amt81=amt81+allamt81; if M81=-3 then amt81=-3; end;
        end;
        if UM81 ge 0 and UI>0 then do; UM81=UM81+1; end;
        if UI>1 then UM81=-3;
        if -3 le uiamt le -1 then do; un81=un81+1; uamt81=uiamt; end;
        if uiamt ge 0 then do; ua81=ua81+1; if ua81=1 and un81=0 then UAMT81=UIAMT;
            if ua81>1 and un81=0 then uamt81=uamt81+uiamt; if UM81=-3 then uamt81=-3; end;
        if WM81 ge 0 and WC>0 then do; WM81=WM81+1; end;
        if WC>1 then WM81=-3;
        if -3 le wcamt le -1 then do; wn81=wn81+1; camt81=wcamt; end;
        if wcamt ge 0 then do; wa81=wa81+1; if wa81=1 and wn81=0 then cAMT81=WCAMT;
            if wa81>1 and wn81=0 then camt81=camt81+wcamt; if WM81=-3 then camt81=-3; end;
    end;

if 25 le L le 36 then do;
    if M82 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M82=M82+1; end;
    if a>1 then M82=-3;      if w>1 then M82=-3;      if f>1 then M82=-3;      if o>1 then M82=-3;
    array allamt82 aamt wamt famt oamt;
    do over allamt82;
        if -3 le allamt82 le -1 then do; nn82=nn82+1; amt82=allamt82; end;
        if allamt82 ge 0 then do; na82=na82+1; if na82=0 and nn82=0 then amt82=allamt82;
            if na82>1 and nn82=0 then amt82=amt82+allamt82; if M82=-3 then amt82=-3; end;
        end;
        if UM82 ge 0 and UI>0 then do; UM82=UM82+1; end;
        if UI>1 then UM82=-3;
        if -3 le uiamt le -1 then do; un82=un82+1; uamt82=uiamt; end;
        if uiamt ge 0 then do; ua82=ua82+1; if ua82=1 and un82=0 then UAMT82=UIAMT;
            if ua82>1 and un82=0 then uamt82=uamt82+uiamt; if UM82=-3 then uamt82=-3; end;
        if WM82 ge 0 and WC>0 then do; WM82=WM82+1; end;
        if WC>1 then WM82=-3;
        if -3 le wcamt le -1 then do; wn82=wn82+1; camt82=wcamt; end;
        if wcamt ge 0 then do; wa82=wa82+1; if wa82=1 and wn82=0 then cAMT82=WCAMT;
            if wa82>1 and wn82=0 then camt82=camt82+wcamt; if WM82=-3 then camt82=-3; end;
    end;

if 37 le L le 48 then do;
    if M83 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M83=M83+1; end;
    if a>1 then M83=-3;      if w>1 then M83=-3;      if f>1 then M83=-3;      if o>1 then M83=-3;
    array allamt83 aamt wamt famt oamt;
    do over allamt83;
        if -3 le allamt83 le -1 then do; nn83=nn83+1; amt83=allamt83; end;
        if allamt83 ge 0 then do; na83=na83+1; if na83=0 and nn83=0 then amt83=allamt83;
            if na83>1 and nn83=0 then amt83=amt83+allamt83; if M83=-3 then amt83=-3; end;
    end;

```

```

if UM83 ge 0 and UI>0 then do; UM83=UM83+1; end;
if UI>1 then UM83=-3;
if -3 le uiamt le -1 then do; un83=un83+1; uamt83=uiamt; end;
if uiamt ge 0 then do; ua83=ua83+1; if ua83=1 and un83=0 then UAMT83=UIAMT;
    if ua83>1 and un83=0 then uamt83=uamt83+uiamt; if UM83=-3 then uamt83=-3; end;
if WM83 ge 0 and WC>0 then do; WM83=WM83+1; end;
if WC>1 then WM83=-3;
if -3 le wcamt le -1 then do; wn83=wn83+1; camt83=wcamt; end;
if wcamt ge 0 then do; wa83=wa83+1; if wa83=1 and wn83=0 then cAMT83=WCAMT;
    if wa83>1 and wn83=0 then camt83=camt83+wcamt; if WM83=-3 then camt83=-3; end;
end;

if 49 le L le 60 then do;
if M84 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M84=M84+1; end;
if a>1 then M84=-3;      if w>1 then M84=-3;      if f>1 then M84=-3;      if o>1 then M84=-3;
array allamt84 aamt wamt famt oamt;
do over allamt84;
    if -3 le allamt84 le -1 then do; nn84=nn84+1; amt84=allamt84; end;
    if allamt84 ge 0 then do; na84=na84+1; if na84=0 and nn84=0 then amt84=allamt84;
        if na84>1 and nn84=0 then amt84=amt84+allamt84; if M84=-3 then amt84=-3; end;
end;
if UM84 ge 0 and UI>0 then do; UM84=UM84+1; end;
if UI>1 then UM84=-3;
if -3 le uiamt le -1 then do; un84=un84+1; uamt84=uiamt; end;
if uiamt ge 0 then do; ua84=ua84+1; if ua84=1 and un84=0 then UAMT84=UIAMT;
    if ua84>1 and un84=0 then uamt84=uamt84+uiamt; if UM84=-3 then uamt84=-3; end;
if WM84 ge 0 and WC>0 then do; WM84=WM84+1; end;
if WC>1 then WM84=-3;
if -3 le wcamt le -1 then do; wn84=wn84+1; camt84=wcamt; end;
if wcamt ge 0 then do; wa84=wa84+1; if wa84=1 and wn84=0 then cAMT84=WCAMT;
    if wa84>1 and wn84=0 then camt84=camt84+wcamt; if WM84=-3 then camt84=-3; end;
end;

if 61 le L le 72 then do;
if M85 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M85=M85+1; end;
if a>1 then M85=-3;      if w>1 then M85=-3;      if f>1 then M85=-3;      if o>1 then M85=-3;
array allamt85 aamt wamt famt oamt;
do over allamt85;
    if -3 le allamt85 le -1 then do; nn85=nn85+1; amt85=allamt85; end;
    if allamt85 ge 0 then do; na85=na85+1; if na85=0 and nn85=0 then amt85=allamt85;
        if na85>1 and nn85=0 then amt85=amt85+allamt85; if M85=-3 then amt85=-3; end;
end;
if UM85 ge 0 and UI>0 then do; UM85=UM85+1; end;
if UI>1 then UM85=-3;
if -3 le uiamt le -1 then do; un85=un85+1; uamt85=uiamt; end;
if uiamt ge 0 then do; ua85=ua85+1; if ua85=1 and un85=0 then UAMT85=UIAMT;
    if ua85>1 and un85=0 then uamt85=uamt85+uiamt; if UM85=-3 then uamt85=-3; end;
if WM85 ge 0 and WC>0 then do; WM85=WM85+1; end;
if WC>1 then WM85=-3;
if -3 le wcamt le -1 then do; wn85=wn85+1; camt85=wcamt; end;
if wcamt ge 0 then do; wa85=wa85+1; if wa85=1 and wn85=0 then cAMT85=WCAMT;
    if wa85>1 and wn85=0 then camt85=camt85+wcamt; if WM85=-3 then camt85=-3; end;
end;

if 73 le L le 84 then do;
if M86 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M86=M86+1; end;

```

```

if a>1 then M86=-3;      if w>1 then M86=-3;      if f>1 then M86=-3;      if o>1 then M86=-3;
array allamt86 aamt wamt famt oamt;
do over allamt86;
  if -3 le allamt86 le -1 then do; nn86=nn86+1; amt86=allamt86; end;
  if allamt86 ge 0 then do; na86=na86+1; if na86=0 and nn86=0 then amt86=allamt86;
    if na86>1 and nn86=0 then amt86=amt86+allamt86; if M86=-3 then amt86=-3; end;
  end;
  if UM86 ge 0 and UI>0 then do; UM86=UM86+1; end;
  if UI>1 then UM86=-3;
  if -3 le uiamt le -1 then do; un86=un86+1; uamt86=uiamt; end;
  if uiamt ge 0 then do; ua86=ua86+1; if ua86=1 and un86=0 then UAMT86=UIAMT;
    if ua86>1 and un86=0 then uamt86=uamt86+uiamt; if UM86=-3 then uamt86=-3; end;
  if WM86 ge 0 and WC>0 then do; WM86=WM86+1; end;
  if WC>1 then WM86=-3;
  if -3 le wcamt le -1 then do; wn86=wn86+1; camt86=wcamt; end;
  if wcamt ge 0 then do; wa86=wa86+1; if wa86=1 and wn86=0 then cAMT86=WCAMT;
    if wa86>1 and wn86=0 then camt86=camt86+wcamt; if WM86=-3 then camt86=-3; end;
end;

if 85 le L le 96 then do;
  if M87 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M87=M87+1; end;
  if a>1 then M87=-3;      if w>1 then M87=-3;      if f>1 then M87=-3;      if o>1 then M87=-3;
  array allamt87 aamt wamt famt oamt;
  do over allamt87;
    if -3 le allamt87 le -1 then do; nn87=nn87+1; amt87=allamt87; end;
    if allamt87 ge 0 then do; na87=na87+1; if na87=0 and nn87=0 then amt87=allamt87;
      if na87>1 and nn87=0 then amt87=amt87+allamt87; if M87=-3 then amt87=-3; end;
    end;
    if UM87 ge 0 and UI>0 then do; UM87=UM87+1; end;
    if UI>1 then UM87=-3;
    if -3 le uiamt le -1 then do; un87=un87+1; uamt87=uiamt; end;
    if uiamt ge 0 then do; ua87=ua87+1; if ua87=1 and un87=0 then UAMT87=UIAMT;
      if ua87>1 and un87=0 then uamt87=uamt87+uiamt; if UM87=-3 then uamt87=-3; end;
    if WM87 ge 0 and WC>0 then do; WM87=WM87+1; end;
    if WC>1 then WM87=-3;
    if -3 le wcamt le -1 then do; wn87=wn87+1; camt87=wcamt; end;
    if wcamt ge 0 then do; wa87=wa87+1; if wa87=1 and wn87=0 then cAMT87=WCAMT;
      if wa87>1 and wn87=0 then camt87=camt87+wcamt; if WM87=-3 then camt87=-3; end;
  end;
end;

if 97 le L le 108 then do;
  if M88 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M88=M88+1; end;
  if a>1 then M88=-3;      if w>1 then M88=-3;      if f>1 then M88=-3;      if o>1 then M88=-3;
  array allamt88 aamt wamt famt oamt;
  do over allamt88;
    if -3 le allamt88 le -1 then do; nn88=nn88+1; amt88=allamt88; end;
    if allamt88 ge 0 then do; na88=na88+1; if na88=0 and nn88=0 then amt88=allamt88;
      if na88>1 and nn88=0 then amt88=amt88+allamt88; if M88=-3 then amt88=-3; end;
    end;
    if UM88 ge 0 and UI>0 then do; UM88=UM88+1; end;
    if UI>1 then UM88=-3;
    if -3 le uiamt le -1 then do; un88=un88+1; uamt88=uiamt; end;
    if uiamt ge 0 then do; ua88=ua88+1; if ua88=1 and un88=0 then UAMT88=UIAMT;
      if ua88>1 and un88=0 then uamt88=uamt88+uiamt; if UM88=-3 then uamt88=-3; end;
    if WM88 ge 0 and WC>0 then do; WM88=WM88+1; end;
    if WC>1 then WM88=-3;
  end;
end;

```

```

if -3 le wcamt le -1 then do; wn88=wn88+1; camt88=wcamt; end;
if wcamt ge 0 then do; wa88=wa88+1; if wa88=1 and wn88=0 then cAMT88=WCAMT;
    if wa88>1 and wn88=0 then camt88=camt88+wcamt; if WM88=-3 then camt88=-3; end;
end;

if 109 le L le 120 then do;
    if M89 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M89=M89+1; end;
    if a>1 then M89=-3;      if w>1 then M89=-3;      if f>1 then M89=-3;      if o>1 then M89=-3;
    array allamt89 aamt wamt famt oamt;
    do over allamt89;
        if -3 le allamt89 le -1 then do; nn89=nn89+1; amt89=allamt89; end;
        if allamt89 ge 0 then do; na89=na89+1; if na89=0 and nn89=0 then amt89=allamt89;
            if na89>1 and nn89=0 then amt89=amt89+allamt89; if M89=-3 then amt89=-3; end;
        end;
        if UM89 ge 0 and UI>0 then do; UM89=UM89+1; end;
        if UI>1 then UM89=-3;
        if -3 le uiamt le -1 then do; un89=un89+1; uamt89=uiamt; end;
        if uiamt ge 0 then do; ua89=ua89+1; if ua89=1 and un89=0 then UAMT89=UIAMT;
            if ua89>1 and un89=0 then uamt89=uamt89+uiamt; if UM89=-3 then uamt89=-3; end;
        if WM89 ge 0 and WC>0 then do; WM89=WM89+1; end;
        if WC>1 then WM89=-3;
        if -3 le wcamt le -1 then do; wn89=wn89+1; camt89=wcamt; end;
        if wcamt ge 0 then do; wa89=wa89+1; if wa89=1 and wn89=0 then cAMT89=WCAMT;
            if wa89>1 and wn89=0 then camt89=camt89+wcamt; if WM89=-3 then camt89=-3; end;
    end;

if 121 le L le 132 then do;
    if M90 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M90=M90+1; end;
    if a>1 then M90=-3;      if w>1 then M90=-3;      if f>1 then M90=-3;      if o>1 then M90=-3;
    array allamt90 aamt wamt famt oamt;
    do over allamt90;
        if -3 le allamt90 le -1 then do; nn90=nn90+1; amt90=allamt90; end;
        if allamt90 ge 0 then do; na90=na90+1; if na90=0 and nn90=0 then amt90=allamt90;
            if na90>1 and nn90=0 then amt90=amt90+allamt90; if M90=-3 then amt90=-3; end;
        end;
        if UM90 ge 0 and UI>0 then do; UM90=UM90+1; end;
        if UI>1 then UM90=-3;
        if -3 le uiamt le -1 then do; un90=un90+1; uamt90=uiamt; end;
        if uiamt ge 0 then do; ua90=ua90+1; if ua90=1 and un90=0 then UAMT90=UIAMT;
            if ua90>1 and un90=0 then uamt90=uamt90+uiamt; if UM90=-3 then uamt90=-3; end;
        if WM90 ge 0 and WC>0 then do; WM90=WM90+1; end;
        if WC>1 then WM90=-3;
        if -3 le wcamt le -1 then do; wn90=wn90+1; camt90=wcamt; end;
        if wcamt ge 0 then do; wa90=wa90+1; if wa90=1 and wn90=0 then cAMT90=WCAMT;
            if wa90>1 and wn90=0 then camt90=camt90+wcamt; if WM90=-3 then camt90=-3; end;
    end;

if 133 le L le 144 then do;
    if M91 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M91=M91+1; end;
    if a>1 then M91=-3;      if w>1 then M91=-3;      if f>1 then M91=-3;      if o>1 then M91=-3;
    array allamt91 aamt wamt famt oamt;
    do over allamt91;
        if -3 le allamt91 le -1 then do; nn91=nn91+1; amt91=allamt91; end;
        if allamt91 ge 0 then do; na91=na91+1; if na91=1 and nn91=0 then amt91=allamt91;
            if na91>1 and nn91=0 then amt91=amt91+allamt91; if M91=-3 then amt91=-3; end;
    end;

```

```

if UM91 ge 0 and UI>0 then do; UM91=UM91+1; end;
if UI>1 then UM91=-3;
if -3 le uiamt le -1 then do; un91=un91+1; uamt91=uiamt; end;
if uiamt ge 0 then do; ua91=ua91+1; if ua91=1 and un91=0 then UAMT91=UIAMT;
    if ua91>1 and un91=0 then uamt91=uamt91+uiamt; if UM91=-3 then uamt91=-3; end;
if WM91 ge 0 and WC>0 then do; WM91=WM91+1; end;
if WC>1 then WM91=-3;
if -3 le wcamt le -1 then do; wn91=wn91+1; camt91=wcamt; end;
if wcamt ge 0 then do; wa91=wa91+1; if wa91=1 and wn91=0 then cAMT91=WCAMT;
    if wa91>1 and wn91=0 then camt91=camt91+wcamt; if WM91=-3 then camt91=-3; end;
end;

if 145 le L le 156 then do;
if M92 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M92=M92+1; end;
if a>1 then M92=-3;      if w>1 then M92=-3;      if f>1 then M92=-3;      if o>1 then M92=-3;
array allamt92 aamt wamt famt oamt;
do over allamt92;
    if -3 le allamt92 le -1 then do; nn92=nn92+1; amt92=allamt92; end;
    if allamt92 ge 0 then do; na92=na92+1; if na92=1 and nn92=0 then amt92=allamt92;
        if na92>1 and nn92=0 then amt92=amt92+allamt92; if M92=-3 then amt92=-3; end;
end;
if UM92 ge 0 and UI>0 then do; UM92=UM92+1; end;
if UI>1 then UM92=-3;
if -3 le uiamt le -1 then do; un92=un92+1; uamt92=uiamt; end;
if uiamt ge 0 then do; ua92=ua92+1; if ua92=1 and un92=0 then UAMT92=UIAMT;
    if ua92>1 and un92=0 then uamt92=uamt92+uiamt; if UM92=-3 then uamt92=-3; end;
if WM92 ge 0 and WC>0 then do; WM92=WM92+1; end;
if WC>1 then WM92=-3;
if -3 le wcamt le -1 then do; wn92=wn92+1; camt92=wcamt; end;
if wcamt ge 0 then do; wa92=wa92+1; if wa92=1 and wn92=0 then cAMT92=WCAMT;
    if wa92>1 and wn92=0 then camt92=camt92+wcamt; if WM92=-3 then camt92=-3; end;
end;

if 157 le L le 168 then do;
if M93 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M93=M93+1; end;
if a>1 then M93=-3;      if w>1 then M93=-3;      if f>1 then M93=-3;      if o>1 then M93=-3;
array allamt93 aamt wamt famt oamt;
do over allamt93;
    if -3 le allamt93 le -1 then do; nn93=nn93+1; amt93=allamt93; end;
    if allamt93 ge 0 then do; na93=na93+1; if na93=1 and nn93=0 then amt93=allamt93;
        if na93>1 and nn93=0 then amt93=amt93+allamt93; if M93=-3 then amt93=-3; end;
end;
if UM93 ge 0 and UI>0 then do; UM93=UM93+1; end;
if UI>1 then UM93=-3;
if -3 le uiamt le -1 then do; un93=un93+1; uamt93=uiamt; end;
if uiamt ge 0 then do; ua93=ua93+1; if ua93=1 and un93=0 then UAMT93=UIAMT;
    if ua93>1 and un93=0 then uamt93=uamt93+uiamt; if UM93=-3 then uamt93=-3; end;
if WM93 ge 0 and WC>0 then do; WM93=WM93+1; end;
if WC>1 then WM93=-3;
if -3 le wcamt le -1 then do; wn93=wn93+1; camt93=wcamt; end;
if wcamt ge 0 then do; wa93=wa93+1; if wa93=1 and wn93=0 then cAMT93=WCAMT;
    if wa93>1 and wn93=0 then camt93=camt93+wcamt; if WM93=-3 then camt93=-3; end;
end;

if 169 le L le 180 then do;
if M94 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M94=M94+1; end;

```

```

if a>1 then M94=-3;      if w>1 then M94=-3;      if f>1 then M94=-3;      if o>1 then M94=-3;
array allamt94 aamt wamt famt oamt;
do over allamt94;
  if -3 le allamt94 le -1 then do; nn94=nn94+1; amt94=allamt94; end;
  if allamt94 ge 0 then do; na94=na94+1; if na94=1 and nn94=0 then amt94=allamt94;
    if na94>1 and nn94=0 then amt94=amt94+allamt94; if M94=-3 then amt94=-3; end;
  end;
  if UM94 ge 0 and UI>0 then do; UM94=UM94+1; end;
  if UI>1 then UM94=-3;
  if -3 le uiamt le -1 then do; un94=un94+1; uamt94=uiamt; end;
  if uiamt ge 0 then do; ua94=ua94+1; if ua94=1 and un94=0 then UAMT94=UIAMT;
    if ua94>1 and un94=0 then uamt94=uamt94+uiamt; if UM94=-3 then uamt94=-3; end;
  if WM94 ge 0 and WC>0 then do; WM94=WM94+1; end;
  if WC>1 then WM94=-3;
  if -3 le wcamt le -1 then do; wn94=wn94+1; camt94=wcamt; end;
  if wcamt ge 0 then do; wa94=wa94+1; if wa94=1 and wn94=0 then cAMT94=WCAMT;
    if wa94>1 and wn94=0 then camt94=camt94+wcamt; if WM94=-3 then camt94=-3; end;
end;

if 181 le L le 192 then do;
  if M95 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M95=M95+1; end;
  if a>1 then M95=-3;      if w>1 then M95=-3;      if f>1 then M95=-3;      if o>1 then M95=-3;
  array allamt95 aamt wamt famt oamt;
  do over allamt95;
    if -3 le allamt95 le -1 then do; nn95=nn95+1; amt95=allamt95; end;
    if allamt95 ge 0 then do; na95=na95+1; if na95=1 and nn95=0 then amt95=allamt95;
      if na95>1 and nn95=0 then amt95=amt95+allamt95; if M95=-3 then amt95=-3; end;
    end;
    if UM95 ge 0 and UI>0 then do; UM95=UM95+1; end;
    if UI>1 then UM95=-3;
    if -3 le uiamt le -1 then do; un95=un95+1; uamt95=uiamt; end;
    if uiamt ge 0 then do; ua95=ua95+1; if ua95=1 and un95=0 then UAMT95=UIAMT;
      if ua95>1 and un95=0 then uamt95=uamt95+uiamt; if UM95=-3 then uamt95=-3; end;
    if WM95 ge 0 and WC>0 then do; WM95=WM95+1; end;
    if WC>1 then WM95=-3;
    if -3 le wcamt le -1 then do; wn95=wn95+1; camt95=wcamt; end;
    if wcamt ge 0 then do; wa95=wa95+1; if wa95=1 and wn95=0 then cAMT95=WCAMT;
      if wa95>1 and wn95=0 then camt95=camt95+wcamt; if WM95=-3 then camt95=-3; end;
  end;

if 193 le L le 204 then do;
  if M96 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M96=M96+1; end;
  if a>1 then M96=-3;      if w>1 then M96=-3;      if f>1 then M96=-3;      if o>1 then M96=-3;
  array allamt96 aamt wamt famt oamt;
  do over allamt96;
    if -3 le allamt96 le -1 then do; nn96=nn96+1; amt96=allamt96; end;
    if allamt96 ge 0 then do; na96=na96+1; if na96=1 and nn96=0 then amt96=allamt96;
      if na96>1 and nn96=0 then amt96=amt96+allamt96; if M96=-3 then amt96=-3; end;
    end;
    if UM96 ge 0 and UI>0 then do; UM96=UM96+1; end;
    if UI>1 then UM96=-3;
    if -3 le uiamt le -1 then do; un96=un96+1; uamt96=uiamt; end;
    if uiamt ge 0 then do; ua96=ua96+1; if ua96=1 and un96=0 then UAMT96=UIAMT;
      if ua96>1 and un96=0 then uamt96=uamt96+uiamt; if UM96=-3 then uamt96=-3; end;
    if WM96 ge 0 and WC>0 then do; WM96=WM96+1; end;
    if WC>1 then WM96=-3;
  end;

```

```

if -3 le wcamt le -1 then do; wn96=wn96+1; camt96=wcamt; end;
  if wcamt ge 0 then do; wa96=wa96+1; if wa96=1 and wn96=0 then cAMT96=WCAMT;
    if wa96>1 and wn96=0 then camt96=camt96+wcamt; if WM96=-3 then camt96=-3; end;
end;

if 205 le L le 216 then do;
  if M97 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M97=M97+1; end;
  if a>1 then M97=-3;      if w>1 then M97=-3;      if f>1 then M97=-3;      if o>1 then M97=-3;
  array allamt97 aamt wamt famt oamt;
  do over allamt97;
    if -3 le allamt97 le -1 then do; nn97=nn97+1; amt97=allamt97; end;
    if allamt97 ge 0 then do; na97=na97+1; if na97=1 and nn97=0 then amt97=allamt97;
      if na97>1 and nn97=0 then amt97=amt97+allamt97; if M97=-3 then amt97=-3; end;
    end;
    if UM97 ge 0 and UI>0 then do; UM97=UM97+1; end;
    if UI>1 then UM97=-3;
    if -3 le uiamt le -1 then do; un97=un97+1; uamt97=uiamt; end;
    if uiamt ge 0 then do; ua97=ua97+1; if ua97=1 and un97=0 then UAMT97=UIAMT;
      if ua97>1 and un97=0 then uamt97=uamt97+uiamt; if UM97=-3 then uamt97=-3; end;
    if WM97 ge 0 and WC>0 then do; WM97=WM97+1; end;
    if WC>1 then WM97=-3;
    if -3 le wcamt le -1 then do; wn97=wn97+1; camt97=wcamt; end;
    if wcamt ge 0 then do; wa97=wa97+1; if wa97=1 and wn97=0 then cAMT97=WCAMT;
      if wa97>1 and wn97=0 then camt97=camt97+wcamt; if WM97=-3 then camt97=-3; end;
end;

if 217 le L le 228 then do;
  if M98 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M98=M98+1; end;
  if a>1 then M98=-3;      if w>1 then M98=-3;      if f>1 then M98=-3;      if o>1 then M98=-3;
  array allamt98 aamt wamt famt oamt;
  do over allamt98;
    if -3 le allamt98 le -1 then do; nn98=nn98+1; amt98=allamt98; end;
    if allamt98 ge 0 then do; na98=na98+1; if na98=1 and nn98=0 then amt98=allamt98;
      if na98>1 and nn98=0 then amt98=amt98+allamt98; if M98=-3 then amt98=-3; end;
    end;
    if UM98 ge 0 and UI>0 then do; UM98=UM98+1; end;
    if UI>1 then UM98=-3;
    if -3 le uiamt le -1 then do; un98=un98+1; uamt98=uiamt; end;
    if uiamt ge 0 then do; ua98=ua98+1; if ua98=1 and un98=0 then UAMT98=UIAMT;
      if ua98>1 and un98=0 then uamt98=uamt98+uiamt; if UM98=-3 then uamt98=-3; end;
    if WM98 ge 0 and WC>0 then do; WM98=WM98+1; end;
    if WC>1 then WM98=-3;
    if -3 le wcamt le -1 then do; wn98=wn98+1; camt98=wcamt; end;
    if wcamt ge 0 then do; wa98=wa98+1; if wa98=1 and wn98=0 then cAMT98=WCAMT;
      if wa98>1 and wn98=0 then camt98=camt98+wcamt; if WM98=-3 then camt98=-3; end;
end;

if 229 le L le 240 then do;
  if M99 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M99=M99+1; end;
  if a>1 then M99=-3;      if w>1 then M99=-3;      if f>1 then M99=-3;      if o>1 then M99=-3;
  array allamt99 aamt wamt famt oamt;
  do over allamt99;
    if -3 le allamt99 le -1 then do; nn99=nn99+1; amt99=allamt99; end;
    if allamt99 ge 0 then do; na99=na99+1; if na99=1 and nn99=0 then amt99=allamt99;
      if na99>1 and nn99=0 then amt99=amt99+allamt99; if M99=-3 then amt99=-3; end;
    end;

```

```

if UM99 ge 0 and UI>0 then do; UM99=UM99+1; end;
if UI>1 then UM99=-3;
if -3 le uiamt le -1 then do; un99=un99+1; uamt99=uiamt; end;
if uiamt ge 0 then do; ua99=ua99+1; if ua99=1 and un99=0 then UAMT99=UIAMT;
    if ua99>1 and un99=0 then uamt99=uamt99+uiamt; if UM99=-3 then uamt99=-3; end;
if WM99 ge 0 and WC>0 then do; WM99=WM99+1; end;
if WC>1 then WM99=-3;
if -3 le wcamt le -1 then do; wn99=wn99+1; camt99=wcamt; end;
if wcamt ge 0 then do; wa99=wa99+1; if wa99=1 and wn99=0 then cAMT99=WCAMT;
    if wa99>1 and wn99=0 then camt99=camt99+wcamt; if WM99=-3 then camt99=-3; end;
end;

if 241 le L le 252 then do;
if M00 ge 0 and a>0 or w>0 or f>0 or o>0 then do; M00=M00+1; end;
if a>1 then M00=-3;      if w>1 then M00=-3;      if f>1 then M00=-3;      if o>1 then M00=-3;
array allamt00 aamt wamt famt oamt;
do over allamt00;
if -3 le allamt00 le -1 then do; nn00=nn00+1; amt00=allamt00; end;
if allamt00 ge 0 then do; na00=na00+1; if na00=1 and nn00=0 then amt00=allamt00;
    if na00>1 and nn00=0 then amt00=amt00+allamt00; if M00=-3 then amt00=-3; end;
end;
if UM00 ge 0 and UI>0 then do; UM00=UM00+1; end;
if UI>1 then UM00=-3;
if -3 le uiamt le -1 then do; un00=un00+1; uamt00=uiamt; end;
if uiamt ge 0 then do; ua00=ua00+1; if ua00=1 and un00=0 then UAMT00=UIAMT;
    if ua00>1 and un00=0 then uamt00=uamt00+uiamt; if UM00=-3 then uamt00=-3; end;
if WM00 ge 0 and WC>0 then do; WM00=WM00+1; end;
if WC>1 then WM00=-3;
if -3 le wcamt le -1 then do; wn00=wn00+1; camt00=wcamt; end;
if wcamt ge 0 then do; wa00=wa00+1; if wa00=1 and wn00=0 then cAMT00=WCAMT;
    if wa00>1 and wn00=0 then camt00=camt00+wcamt; if WM00=-3 then camt00=-3; end;
end;

*total months - part 2;
if 1 le L le 245 then do;
if TTLM ge 0 and A>0 or W>0 or F>0 or O>0 then do; TTLM=TTLM+1; end;
if a>1 then TTLM=-3;          if w>1 then TTLM=-3;      if f>1 then TTLM=-3;      if o>1 then TTLM=-3;
if TTLU ge 0 and UI>0 then do; TTLU=TTLU+1; end;
if UI>1 then TTLU=-3;
if TTLW ge 0 and WC>0 then do; TTLW=TTLW+1; end;
if WC>1 then TTLW=-3;
end;

**for income program amounts;
if 217 le L le 228 then do;
if w>0 or f>0 then do; out=1; end;
if incprg ge 0 and (a>0 or o>0 or ui>0 or wc>0) then do; incprg=incprg+1; end;
if a>1 or o>1 or ui>1 or wc>1 then incprg=-3;
array prgamts aamt oamt uiamt wcamt;
do over prgamts;
if -3 le prgamts le -1 then do; nnp=nnp+1; prgamt=prgamts; end;
if prgamts ge 0 then do; nap=nap+1; if nap=1 and nnp=0 then prgamt=prgamts;
    if nap>1 and nnp=0 then prgamt=prgamt+prgamts; if incprg=-3 then prgamt=-3; end;
end;
end;
end;

```

```

end;                                /*3. end*/

do i=1 to 21;
  if gprg_ever=. then gprg_ever=0;
  oldM=gtotal;                      oldamt=gatotal;
  if gtotal=-4 then gtotal=0;          if gatotal=-4 then gatotal=0;
  if M ge 0 and gtotal ge 0 then do; gtotal=M+gtotal; end;
  if M<0 or gtotal<0 then do; gtotal=-3; end;
  if amt ge 0 and gatotal ge 0 then do; gatotal=amt+gatotal; end;
  if amt<0 or gatotal<0 or (gtotal<0 and gtotal ne -4) then do; gatotal=-3; end;
  if amt=0 and ckamt=-4 and gtotal=0 then do; gatotal=-4; end;
end;

*total months, part 2;
if ttlm ge 0 and gprg_ever ge 0 then do; ttlm=ttlm+gprg_ever; end;
if ttlm<0 or gprg_ever<0 then do; ttlm=-3; end;
if ttlu ge 0 and ui_ever ge 0 then do; ttlu=ttlu+ui_ever; end;
if ttlu<0 or ui_ever<0 then do; ttlu=-3; end;
if ttlw ge 0 and wc_ever ge 0 then do; ttlw=ttlw+wc_ever; end;
if ttlw<0 or wc_ever<0 then do; ttlw=-3; end;

do i=1 to 21;
  olduiamt=uiatotal;      if uitotal=-4 then uitotal=0;      if uiatotal=-4 then uiatotal=0;
  if UM ge 0 and uitotal ge 0 then do; uitotal=UM+uitotal; end;
  if UM<0 or uitotal<0 then do; uitotal=-3; end;
  if uamt ge 0 and uiatotal ge 0 then do; uiatotal=uamt+uiatotal; end;
  if uamt<0 or uiatotal<0 or uitotal<0 then do; uiatotal=-3; end;
  if uamt=0 and ckuiamt=-4 and uitotal=0 then do; uiatotal=-4; end;
  if uamt=0 and um=0 then do; uiatotal=-4; end;
end;

do i=1 to 21;
  oldwcamt=wcatotal;      if wctotal=-4 then wctotal=0;      if wcatotal=-4 then wcatotal=0;
  if WM ge 0 and wctotal ge 0 then do; wctotal=WM+wctotal; end;
  if WM<0 or wctotal<0 then do; wctotal=-3; end;
  if camt ge 0 and wcatotal ge 0 then do; wcatotal=camt+wcatotal; end;
  if camt<0 or wcatotal<0 or wctotal<0 then do; wcatotal=-3; end;
  if camt=0 and ckwcamt=-4 and wctotal=0 then do; wcatotal=-4; end;
  if camt=0 and wm=0 then do; wcatotal=-4; end;
end;

do i=1 to 21;
  if p1210=-4 then do;
    gtotal=-4;      gatotal=-4;      ttlM=-4;      uitotal=-4;      nu=-4;      uiatotal=-4;
    TTLU=-4;      wctotal=-4;      wcatotal=-4;      TTLW=-4;      incprg=-4;      prgamt=-4;
  end;
  if p1210=-5 then do;
    gtotal=-5;      gatotal=-5;      ttlM=-5;      uitotal=-5;      nu=-5;      uiatotal=-5;
    TTLU=-5;      wctotal=-5;      wcatotal=-5;      TTLW=-5;      incprg=-5;      prgamt=-5;
  end;
end;

endsas;

```